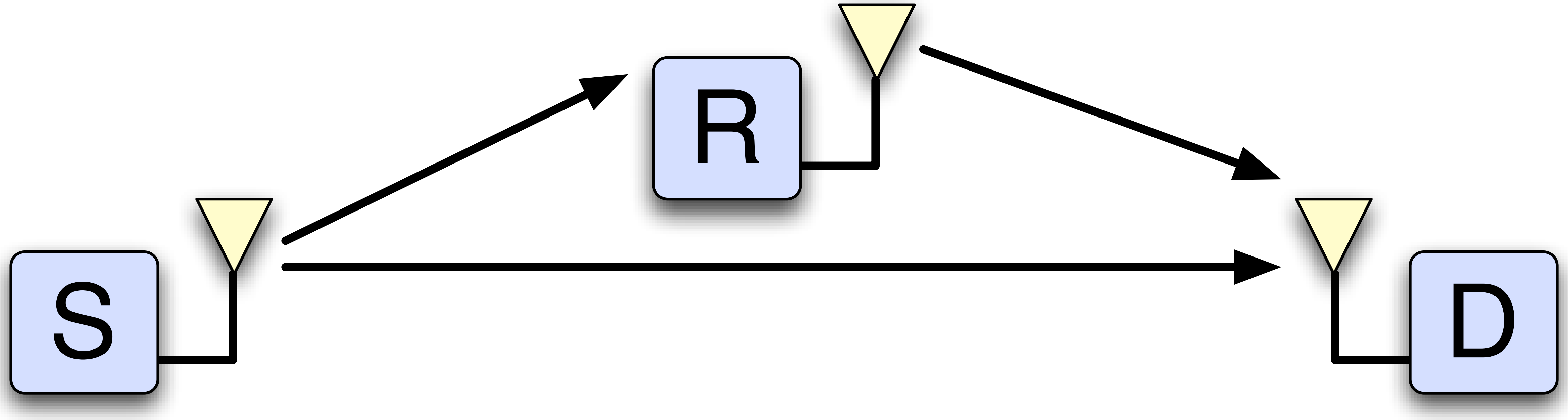
# Design, Implementation and Characterization of a Cooperative Communications System

Patrick Murphy
PhD Defense
November 23, 2010

# The Problem

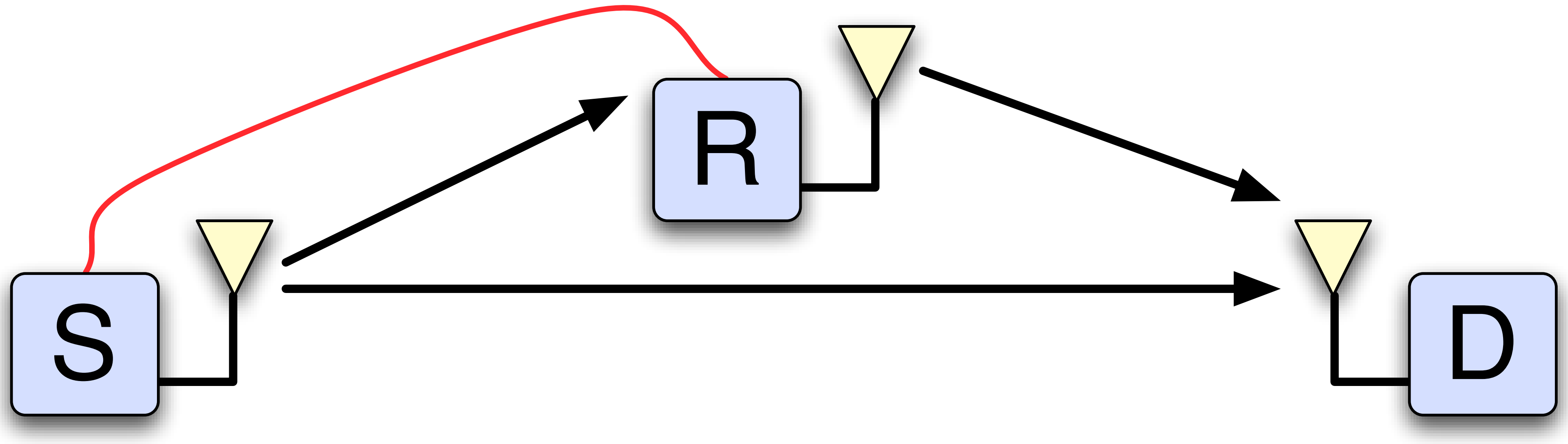
- Wireless always needs to be faster & more reliable
  - Use more power & bandwidth
  - Exploit spatial resources
    - More antennas - throughput and reliability
  - More processing resources
    - Stronger codes
    - Iterative processing beyond codes
  - Exploit *other users*’ resources

# Cooperative Communications

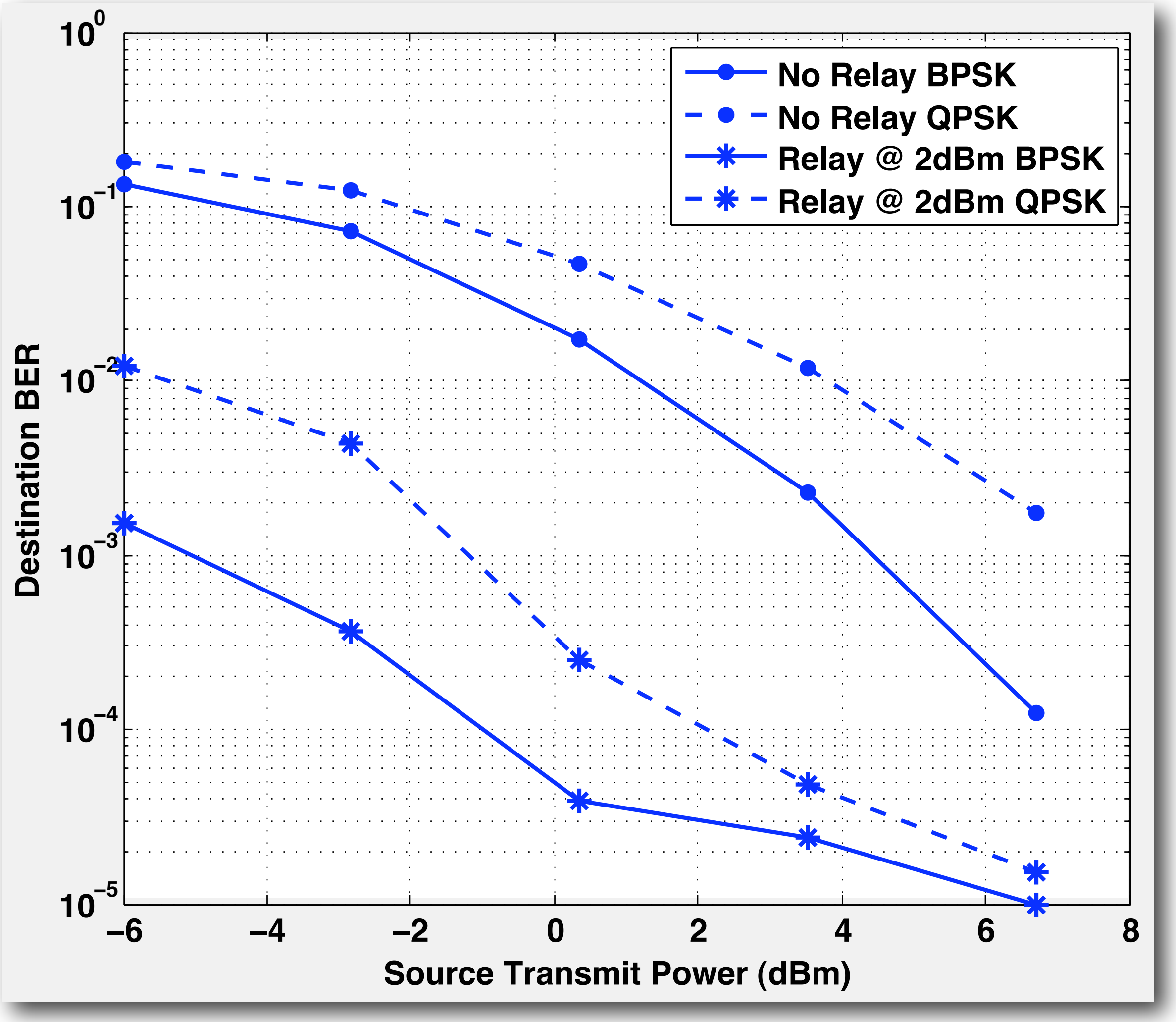


- Cooperation at the physical layer
- Originally described in the 1970's
- Revived in the 1990's
- Lots of promise in theory
- But will it actually work?

# Our Previous Results



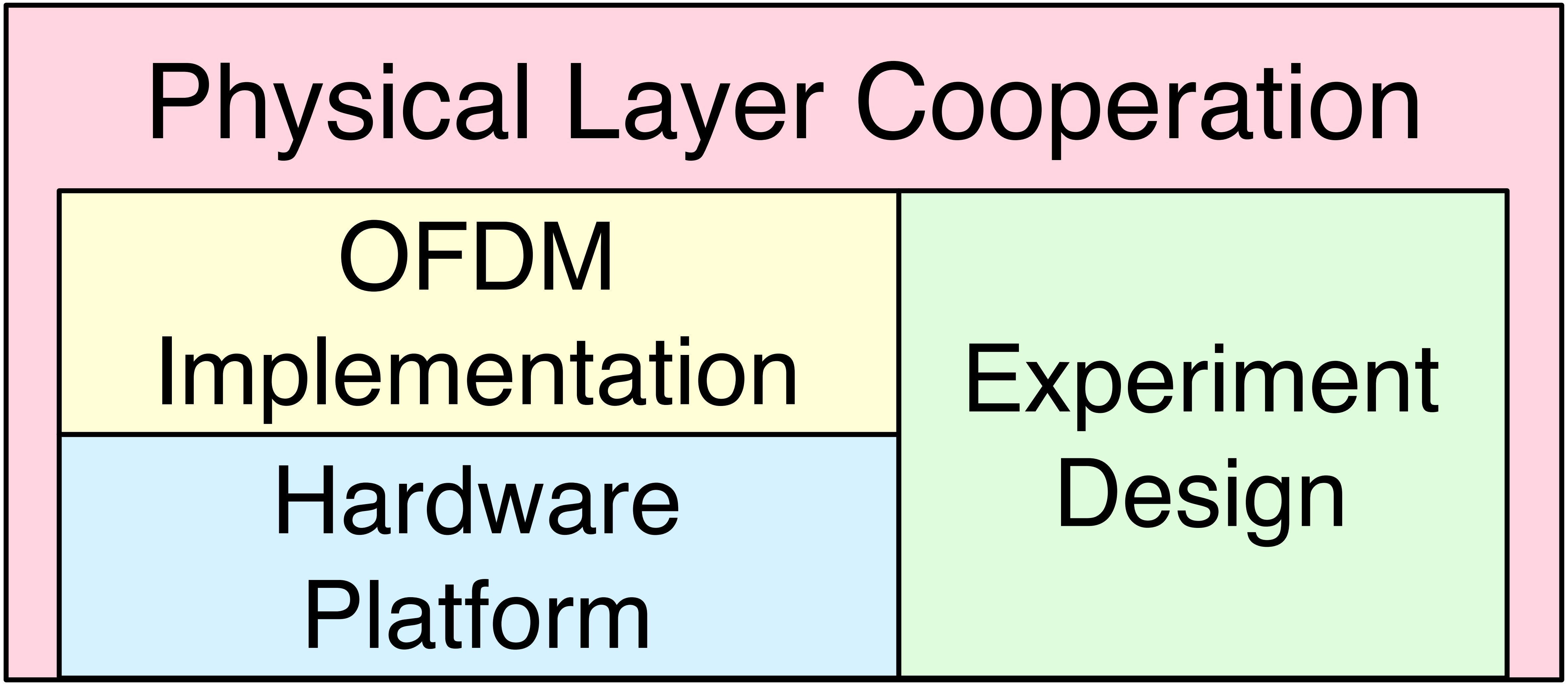
- AF only
- Fixed length transmissions
- Dedicated relay
- Cheated on synchronization
- Partial characterization



# Goals

- Complete real-time cooperative transceiver
  - AF, DF and non-cooperative schemes
  - No cheating
- Rigorous characterization
- Support for MAC protocol development
  - Any role (S/R/D) at every node
  - Cooperative scheme per-packet

# Requirements



Physical Layer Cooperation

OFDM Implementation

Hardware Platform

Experiment Design

# Outline

- Brief background
- Carrier frequency offsets
- Experiment design
- Characterization results
- Future work

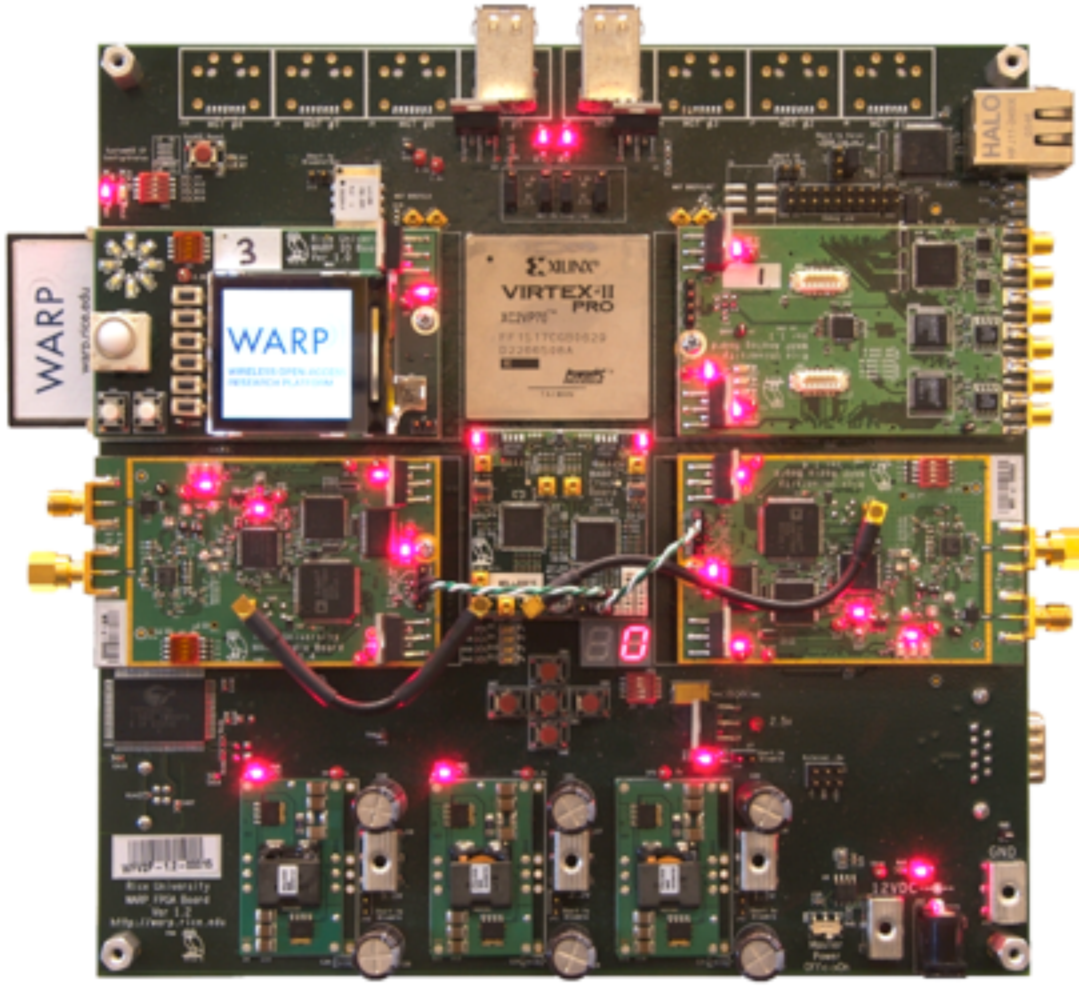
# Background

- Platform selection
- Cooperative schemes
- Transceiver basics

# Wireless Open-Access Research Platform

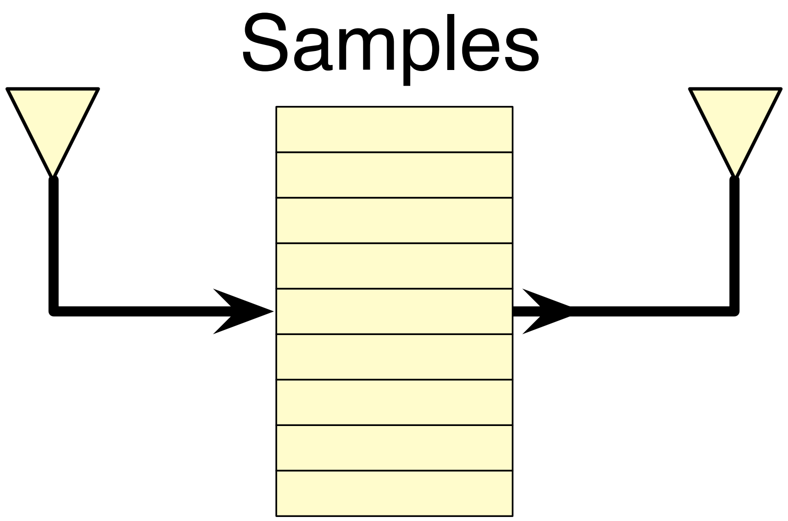
- Rice WARP is obvious choice
- Designed for high-performance wireless prototyping
- Large community of active users
- Local expertise

WARP

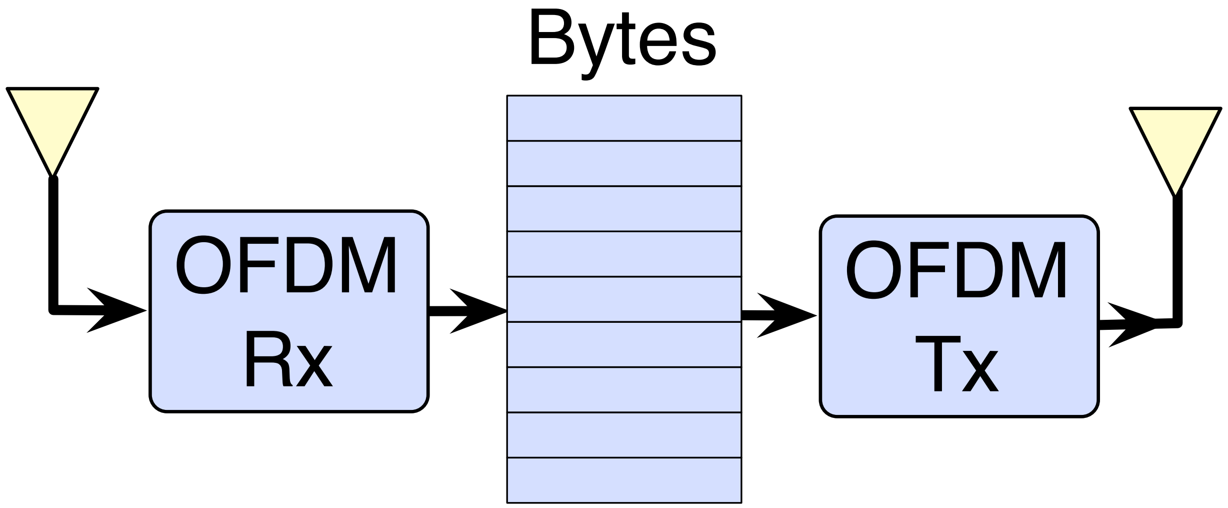


# Cooperative Schemes

- Amplify and Forward
  - Relay captures raw I/Q samples
  - Re-transmits scaled waveform



- Decode and Forward
  - Relay implements full PHY
  - Re-transmits payload



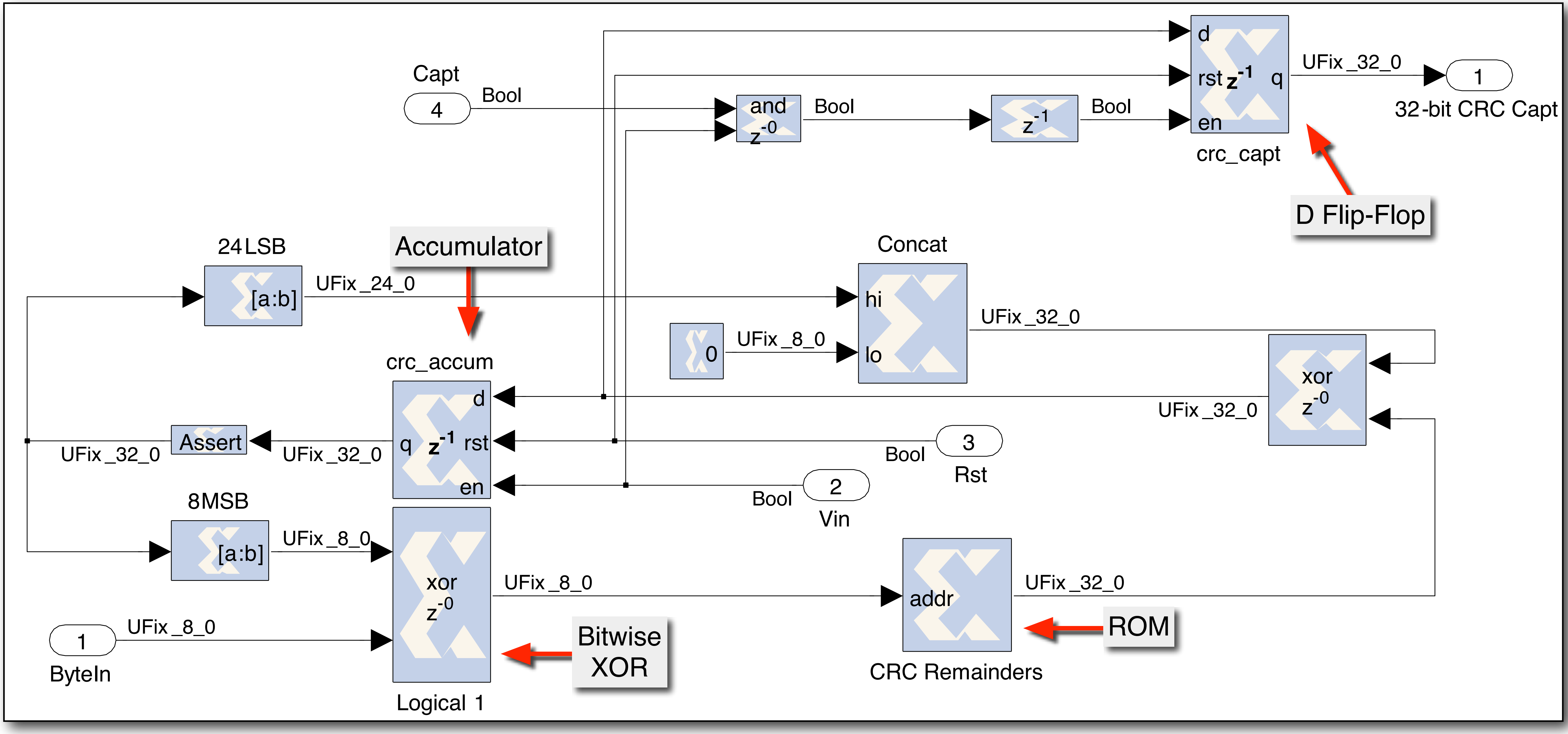
# Transceiver Basics

- How to implement real-time PHY?
- How to go wideband?
- How to orthogonalize transmissions?
- How to synchronize transmissions?

# FPGA Design

- Xilinx System Generator
  - Graphical design entry via Simulink
  - Matching simulation and HDL generation
- Still low-level design
  - Same cores as HDL + Coregen
- One big model implements everything

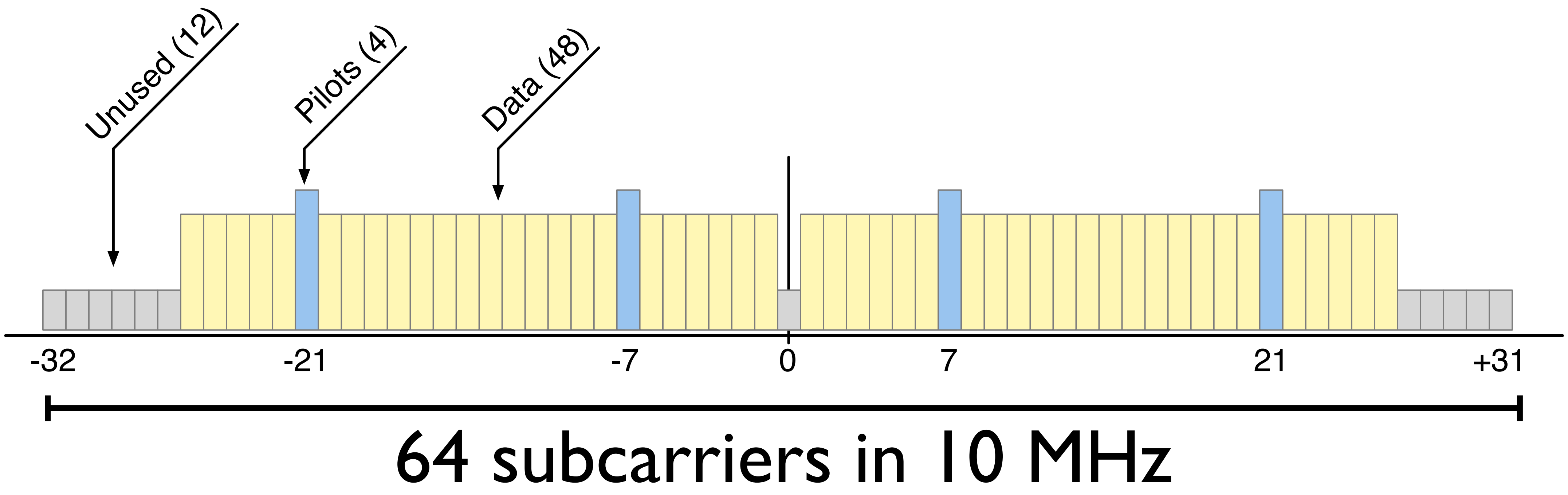
# FPGA Design



Capt
4
Bool
and
z-0
Bool
z-1
Bool
d
rst z-1 q
en
crc_capt
UFix_32_0
1
32-bit CRC Capt
D Flip-Flop
24LSB
[a:b]
UFix_24_0
Accumulator
Concat
hi
lo
0
UFix_8_0
UFix_32_0
xor
z-0
UFix_32_0
crc_accum
d
q z-1 rst
en
Assert
UFix_32_0
UFix_32_0
3
Bool
Rst
2
Bool
Vin
8MSB
[a:b]
UFix_8_0
xor
z-0
UFix_8_0
addr
UFix_32_0
1
UFix_8_0
ByteIn
Logical 1
Bitwise XOR
CRC Remainders
ROM

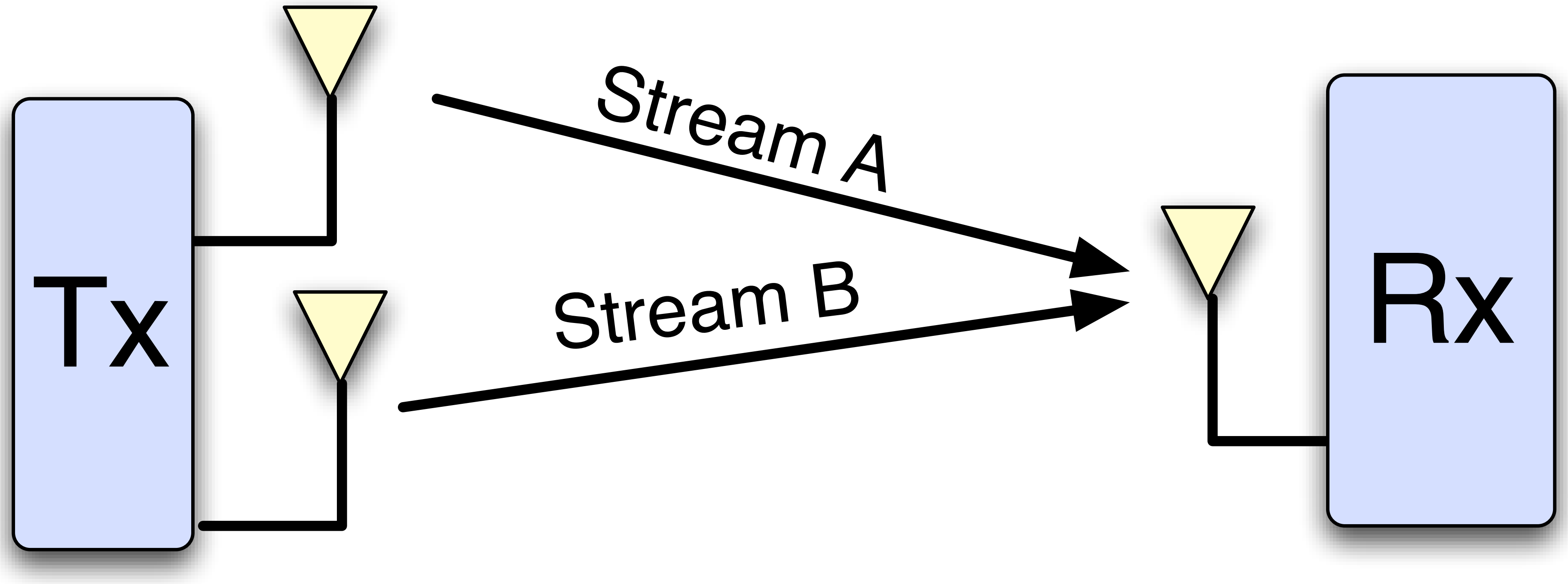
# OFDM

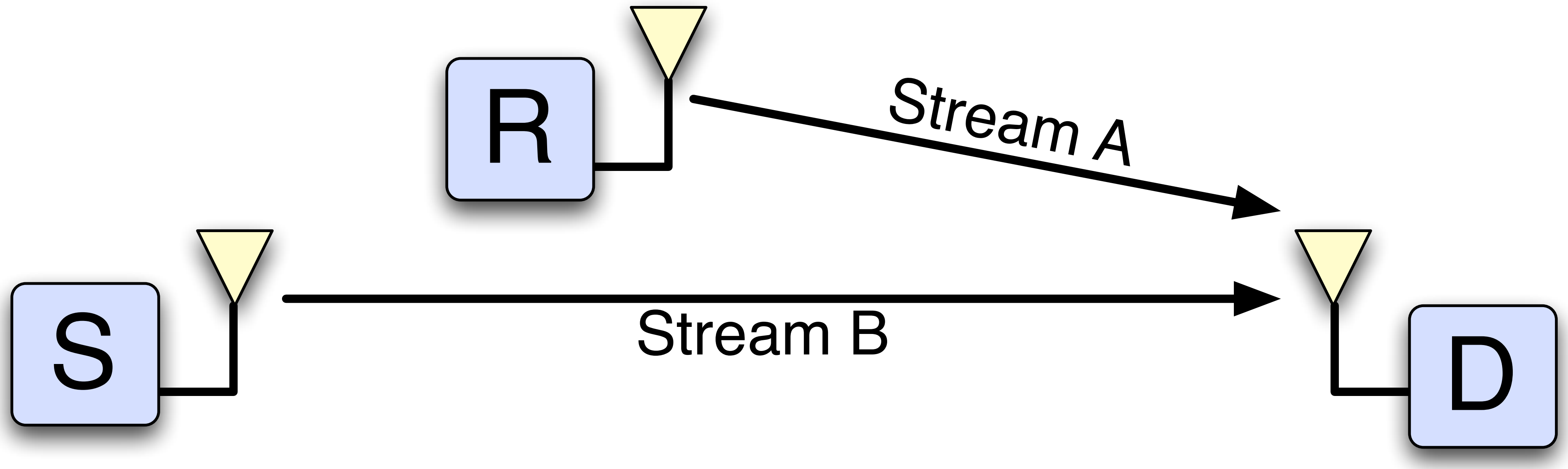
- Orthogonal frequency division multiplexing
  - Foundation for 802.11a/g/n, WiMAX, etc.
  - Good multipath tolerance
  - Eases receiver implementation



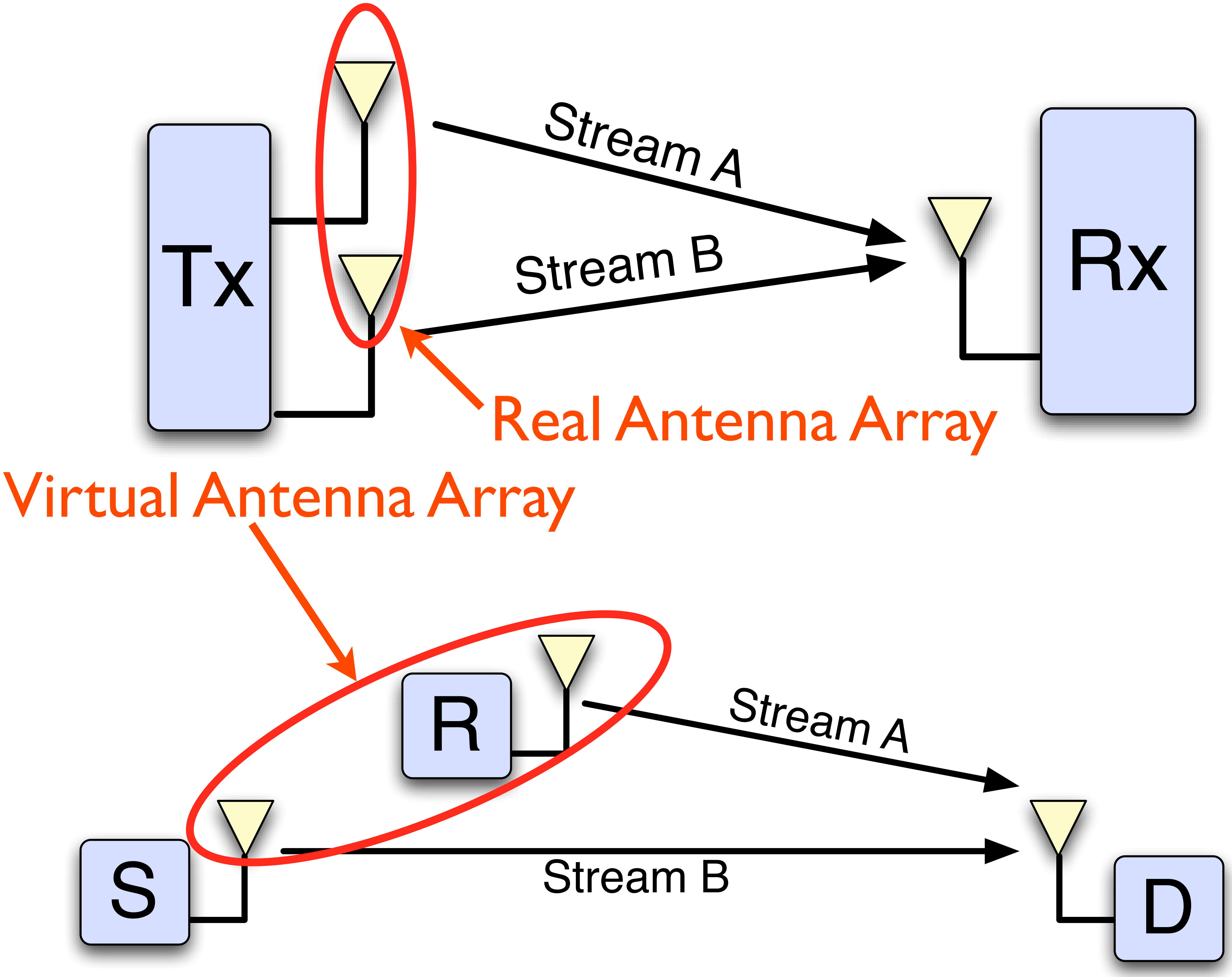
64 subcarriers in 10 MHz

# Distributed Alamouti Code

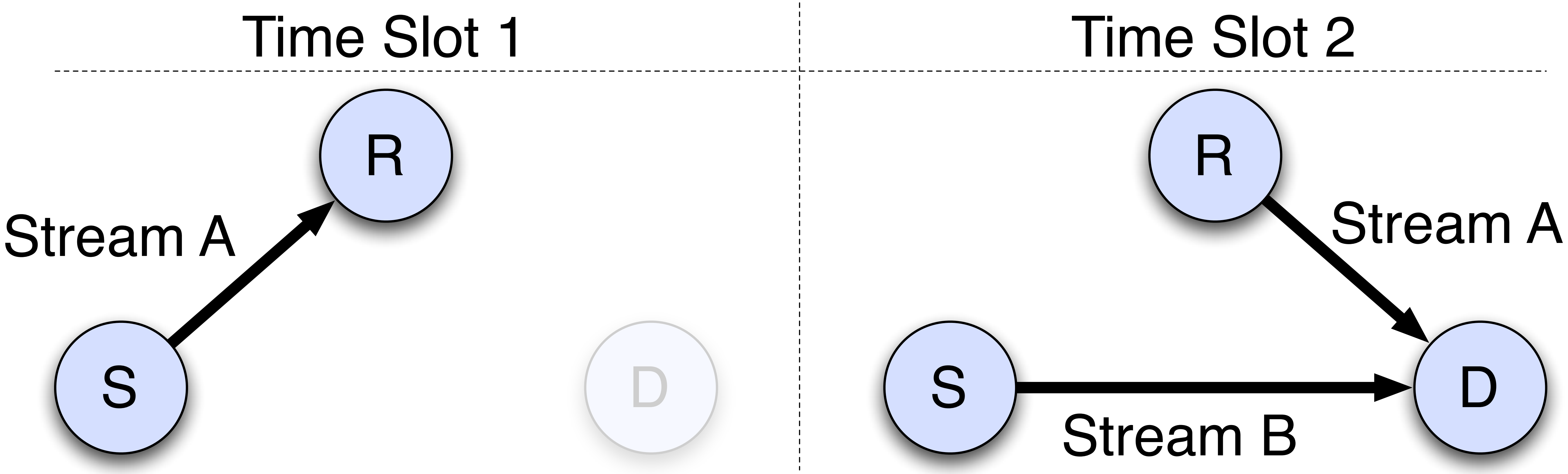




# Distributed Alamouti Code

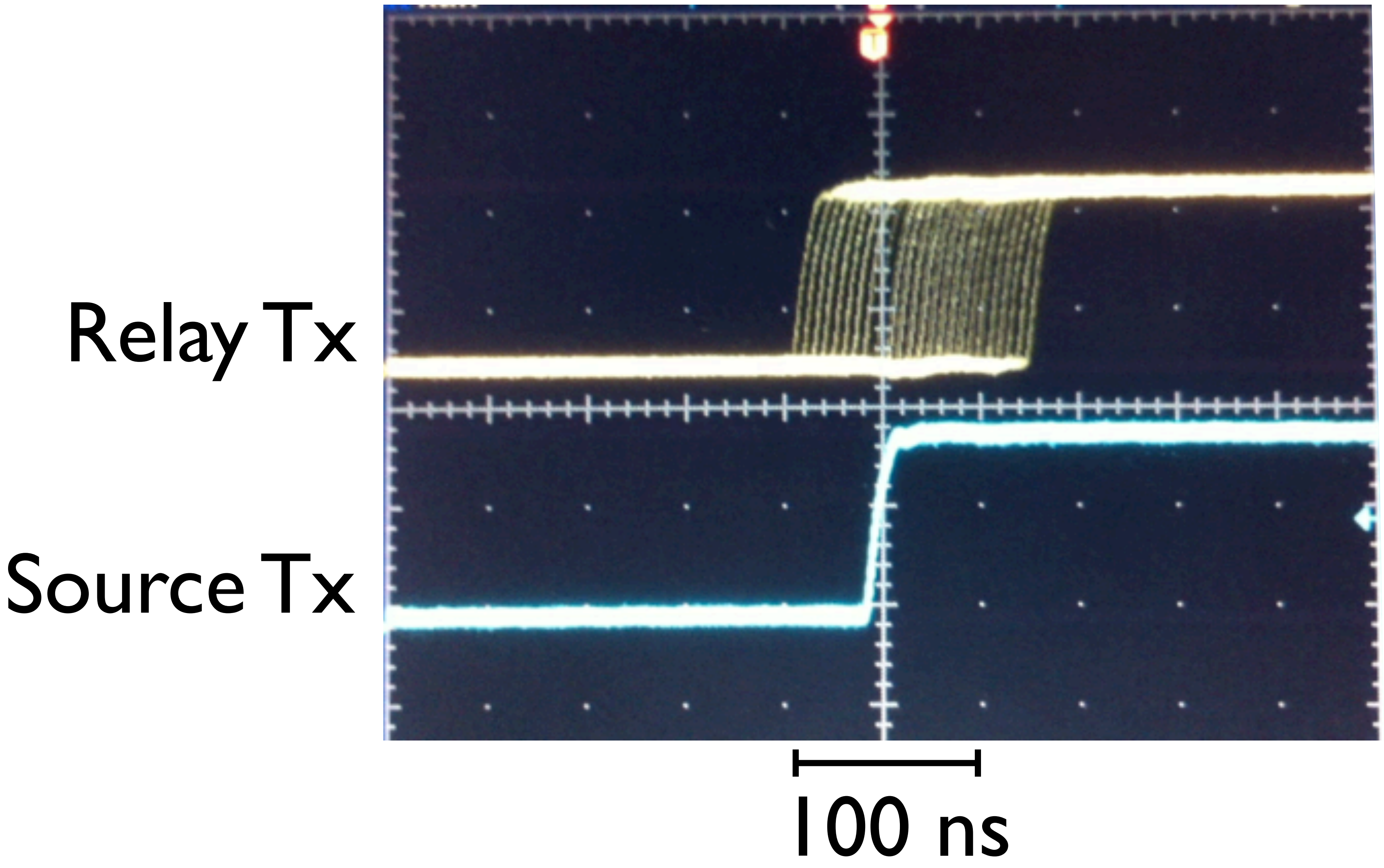


# Distributed Alamouti Code



# Synchronized Transmissions

- Implements Rx/Tx turnaround in FPGA
  - Fast & deterministic latency
  - Programmable for other MAC uses



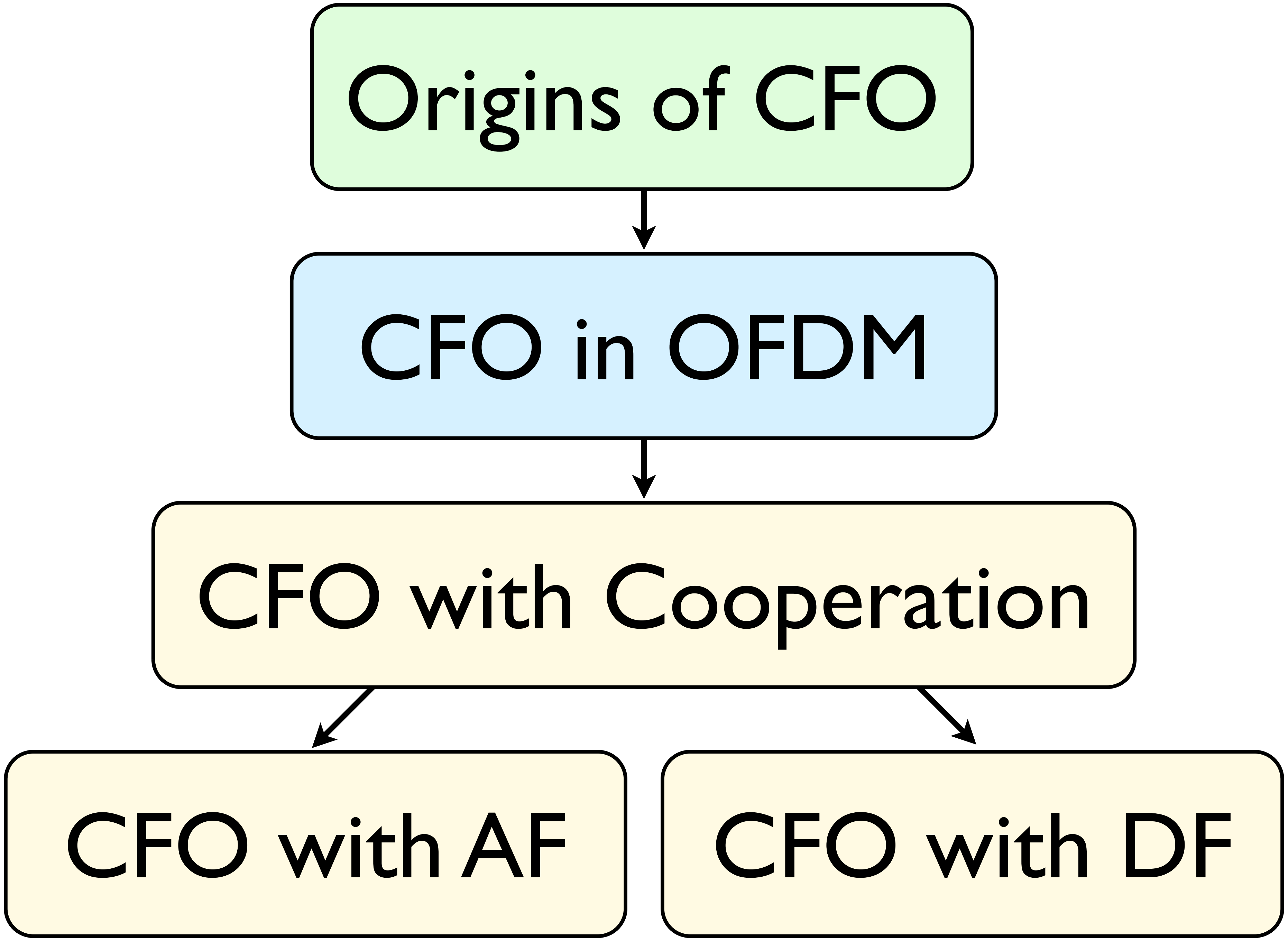
# Transceiver Basics

- How to implement real-time PHY? ✓ FPGA Logic
- How to go wideband? ✓ OFDM
- How to orthogonalize transmissions? ✓ Alamouti STBC
- How to synchronize transmissions? ✓ Rx/Tx turnaround in FPGA

# Outline

- Brief background
- Carrier frequency offsets
- Experiment design
- Characterization results
- Future work

# Carrier Frequency Offset



Origins of CFO

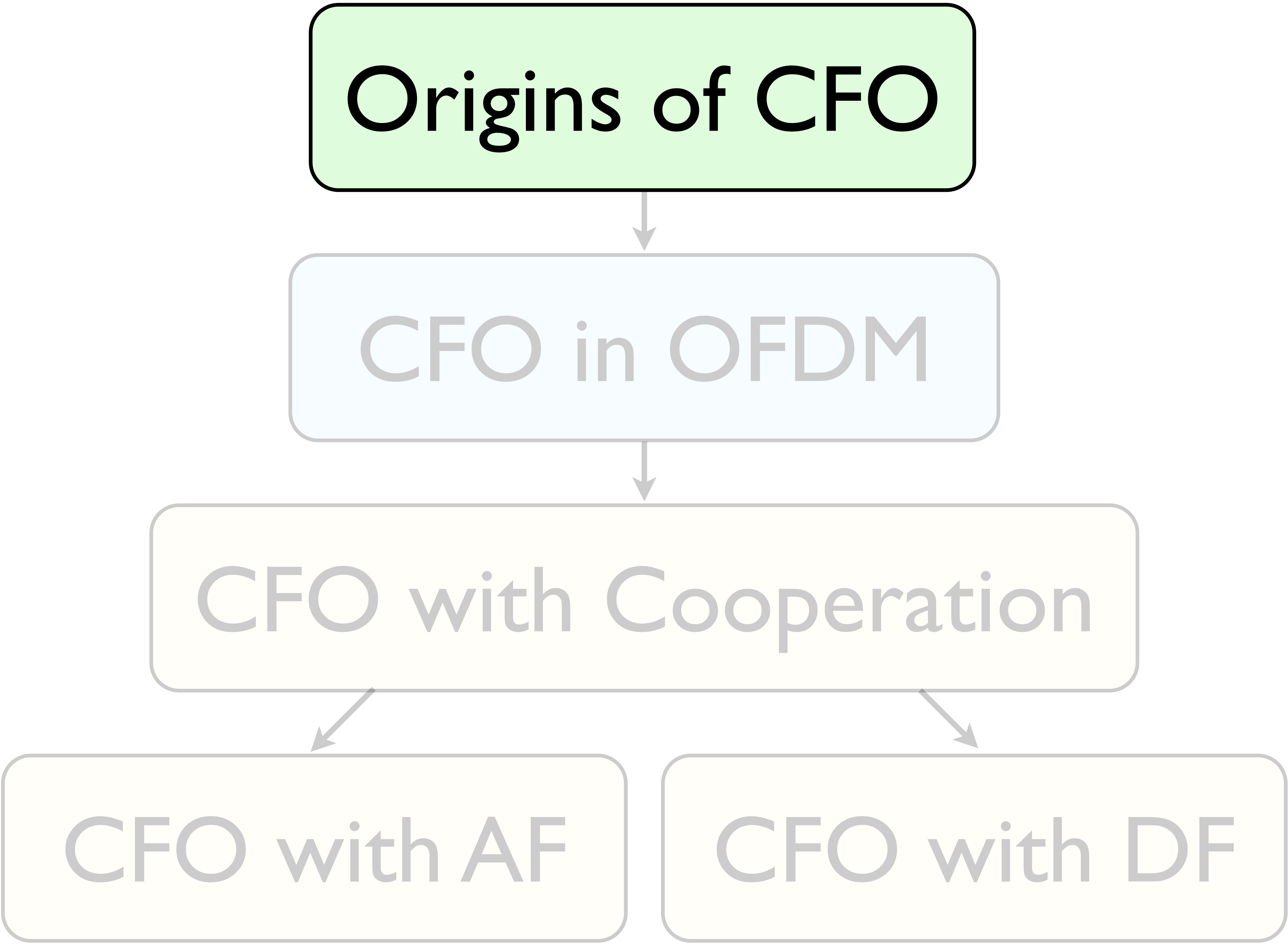
CFO in OFDM

CFO with Cooperation

CFO with AF

CFO with DF

# Carrier Frequency Offset



Origins of CFO

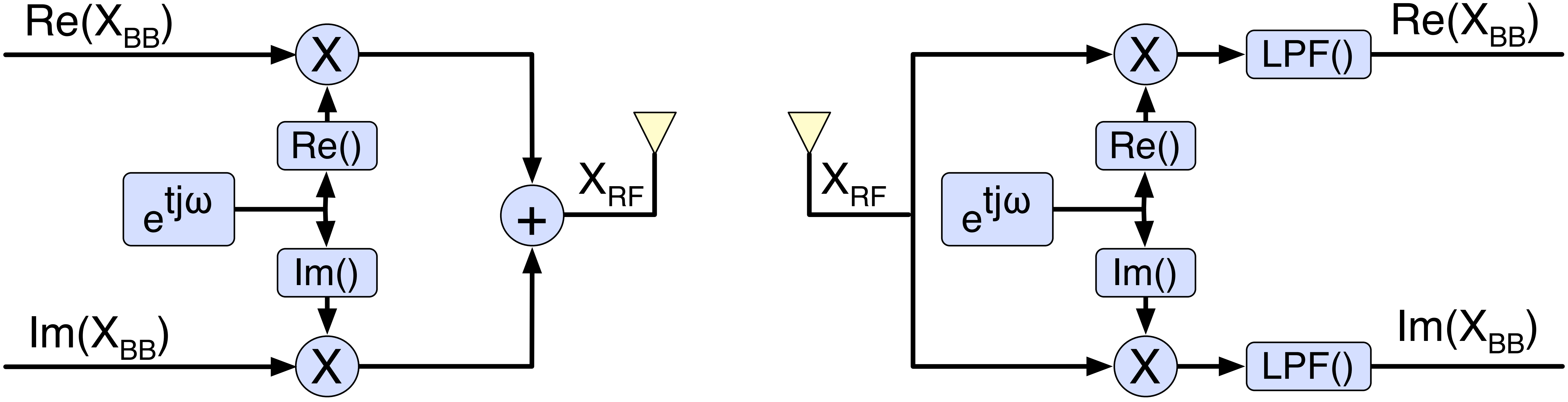
CFO in OFDM

CFO with Cooperation

CFO with AF

CFO with DF

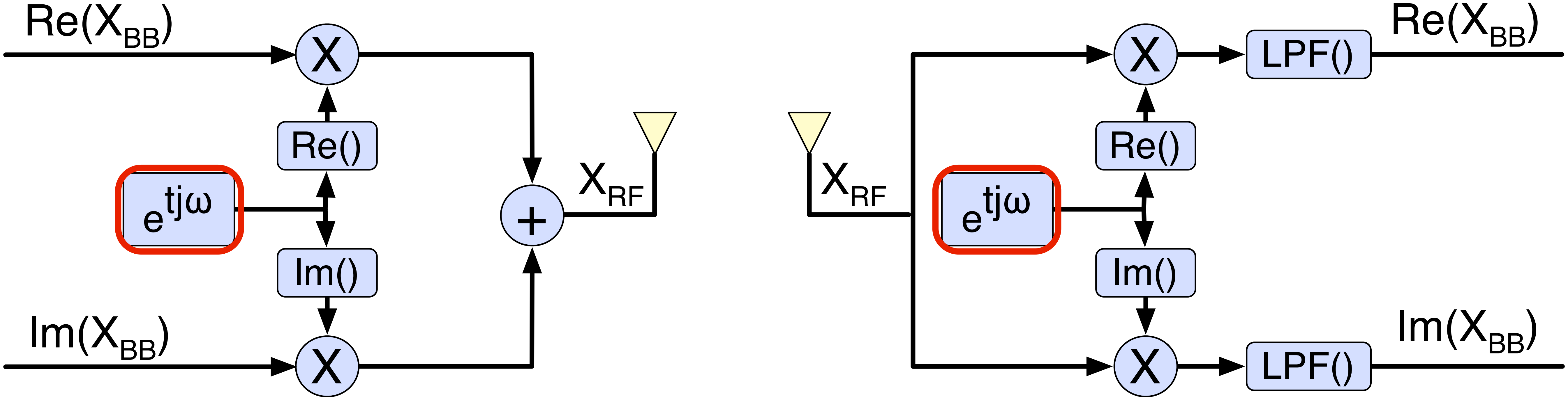
# Carrier Frequency Offset



*RF Upconverter*

*RF Downconverter*

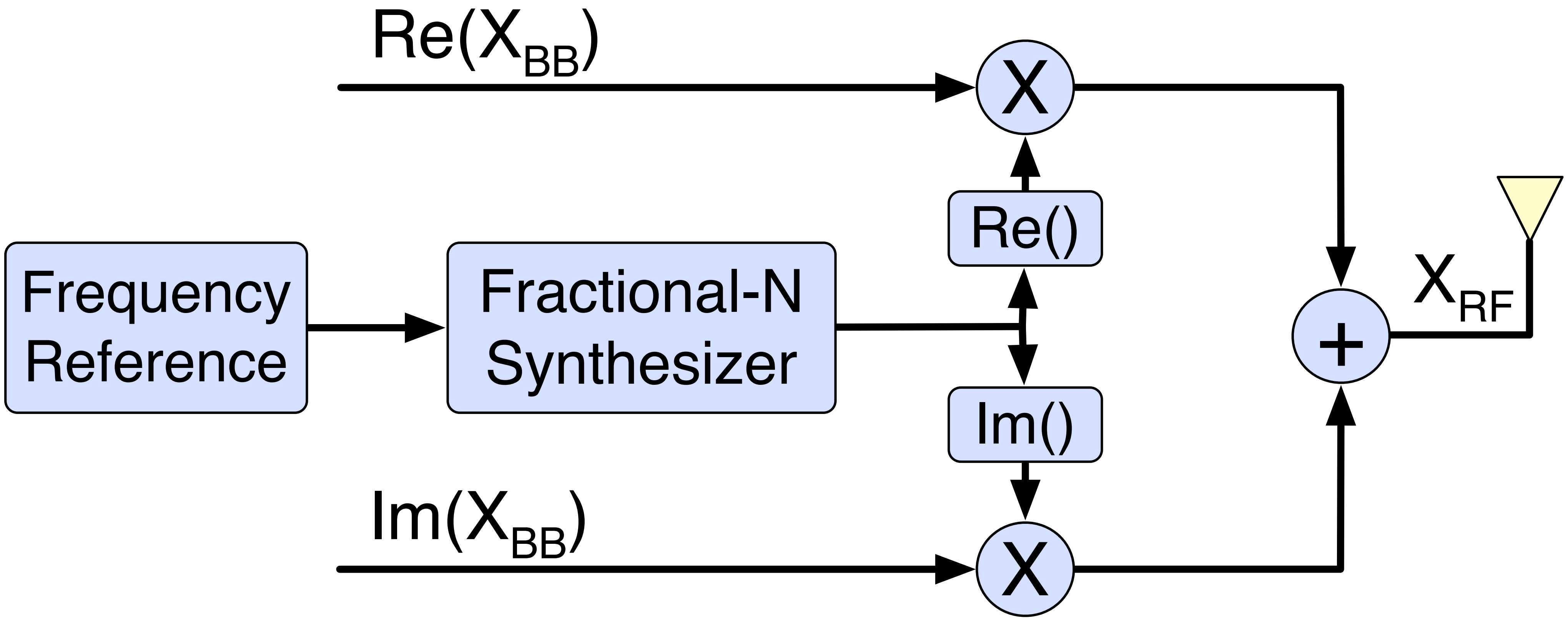
# Carrier Frequency Offset



*RF Upconverter*

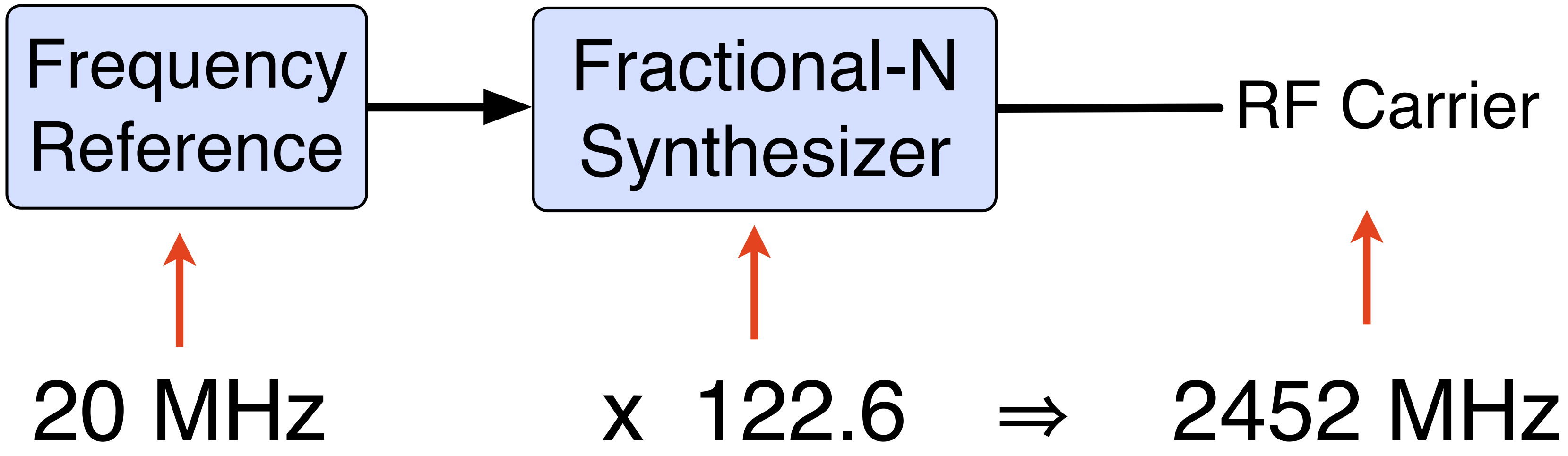
*RF Downconverter*

# Carrier Frequency Offset



CFO characteristics very hardware specific

# Carrier Frequency Offset



Frequency Reference → Fractional-N Synthesizer → RF Carrier

20 MHz x 122.6 ⇒ 2452 MHz

# Carrier Frequency Offset

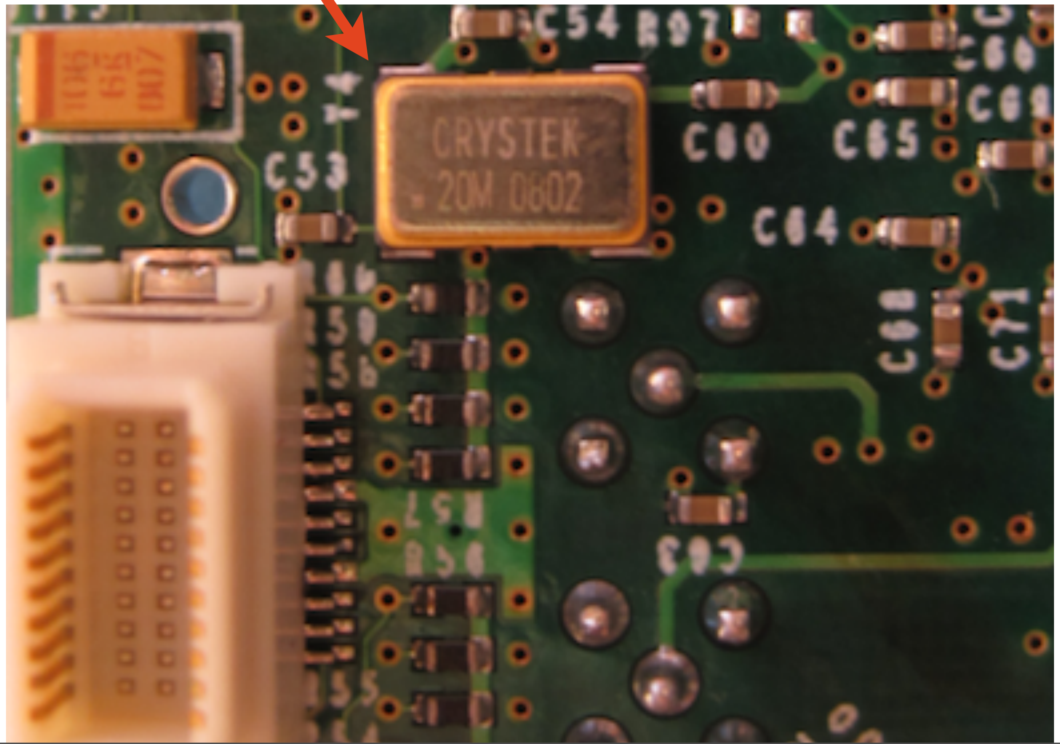
Frequency Reference

20 MHz

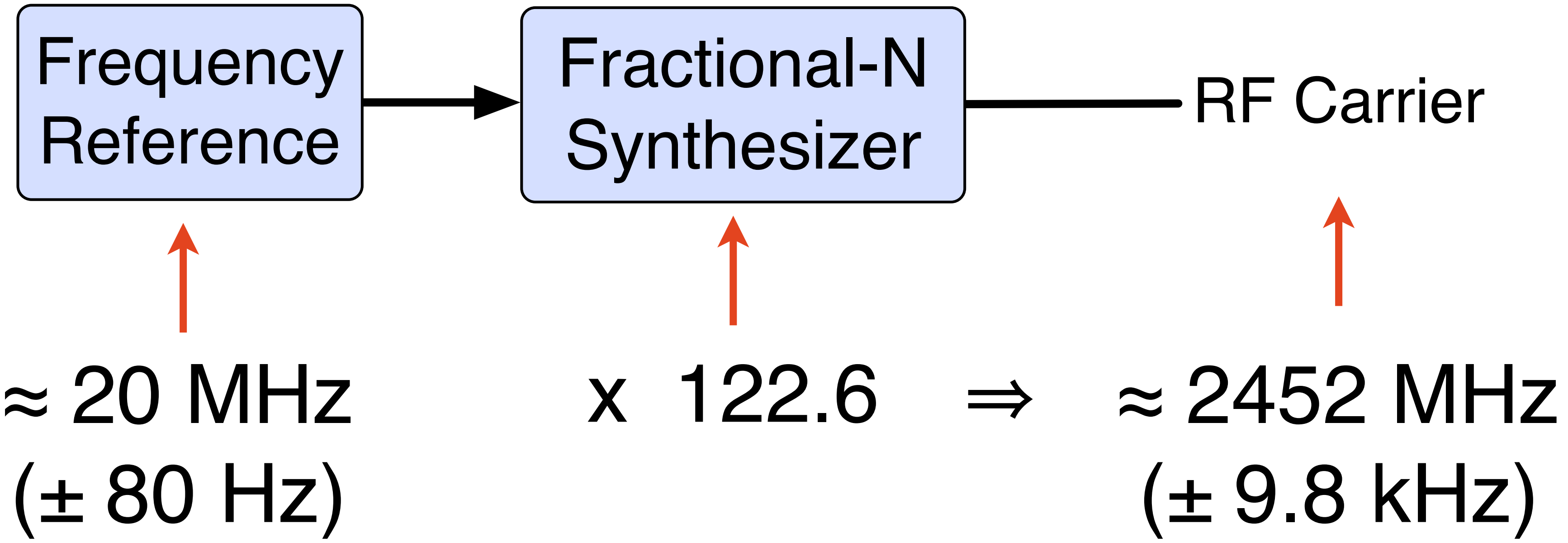
**CVT32 Model**

3.2x5.0 mm SMD, **3V, TCVCXO**

| | |
|---|---|
| **Frequency Range:** | 10MHz to 30MHz |
| **Calibration Tolerance:** | ±1.5ppm |
| **Frequency Stability:** | ±2.5ppm |
| **Temperature Range:** | -20°C to 80°C |



# Carrier Frequency Offset



Frequency Reference

Fractional-N Synthesizer

RF Carrier

≈ 20 MHz (± 80 Hz)

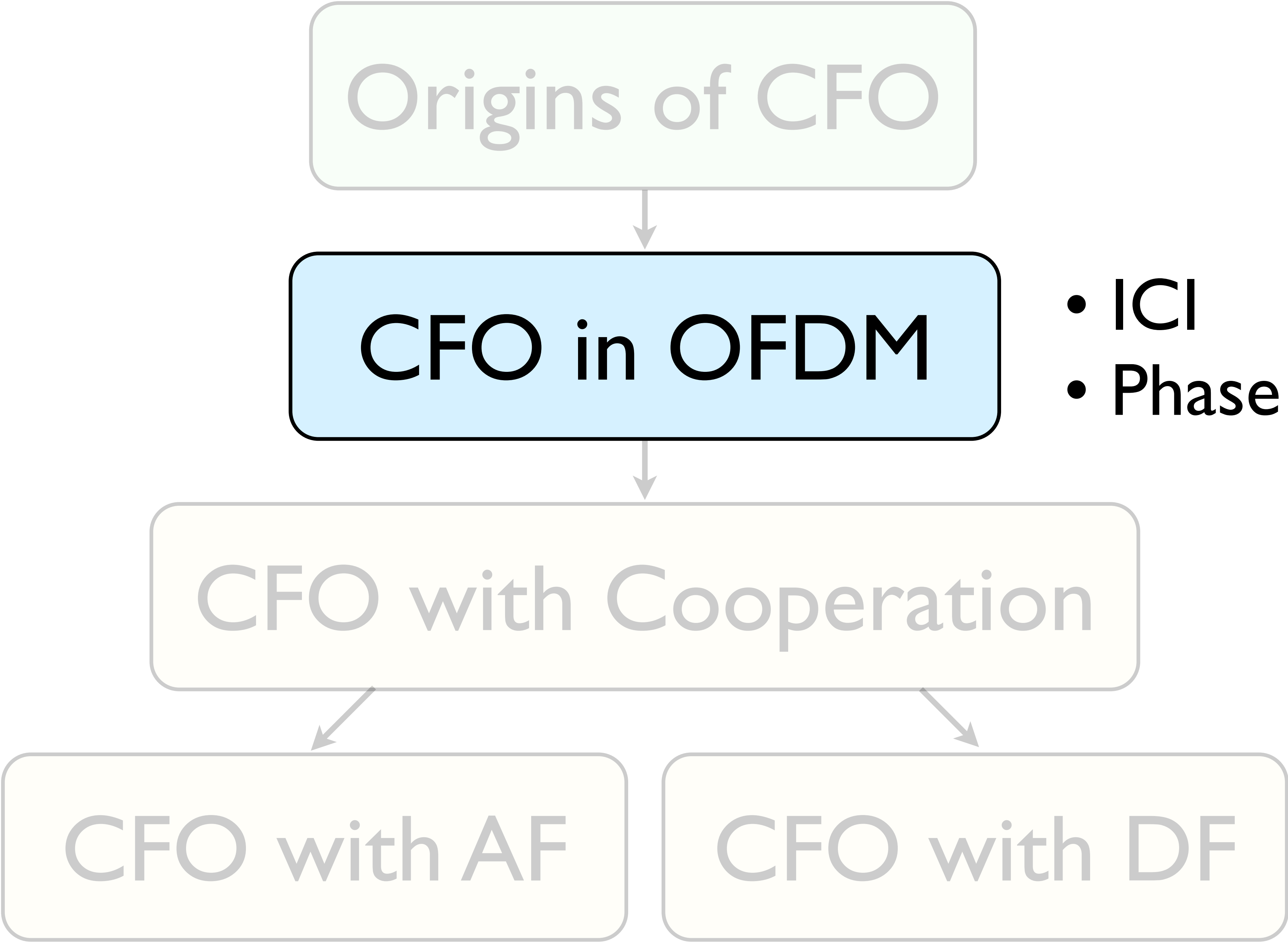
x 122.6

⇒

≈ 2452 MHz (± 9.8 kHz)

⇒Max node-to-node CFO ≈20 kHz

# Carrier Frequency Offset



Origins of CFO

CFO in OFDM

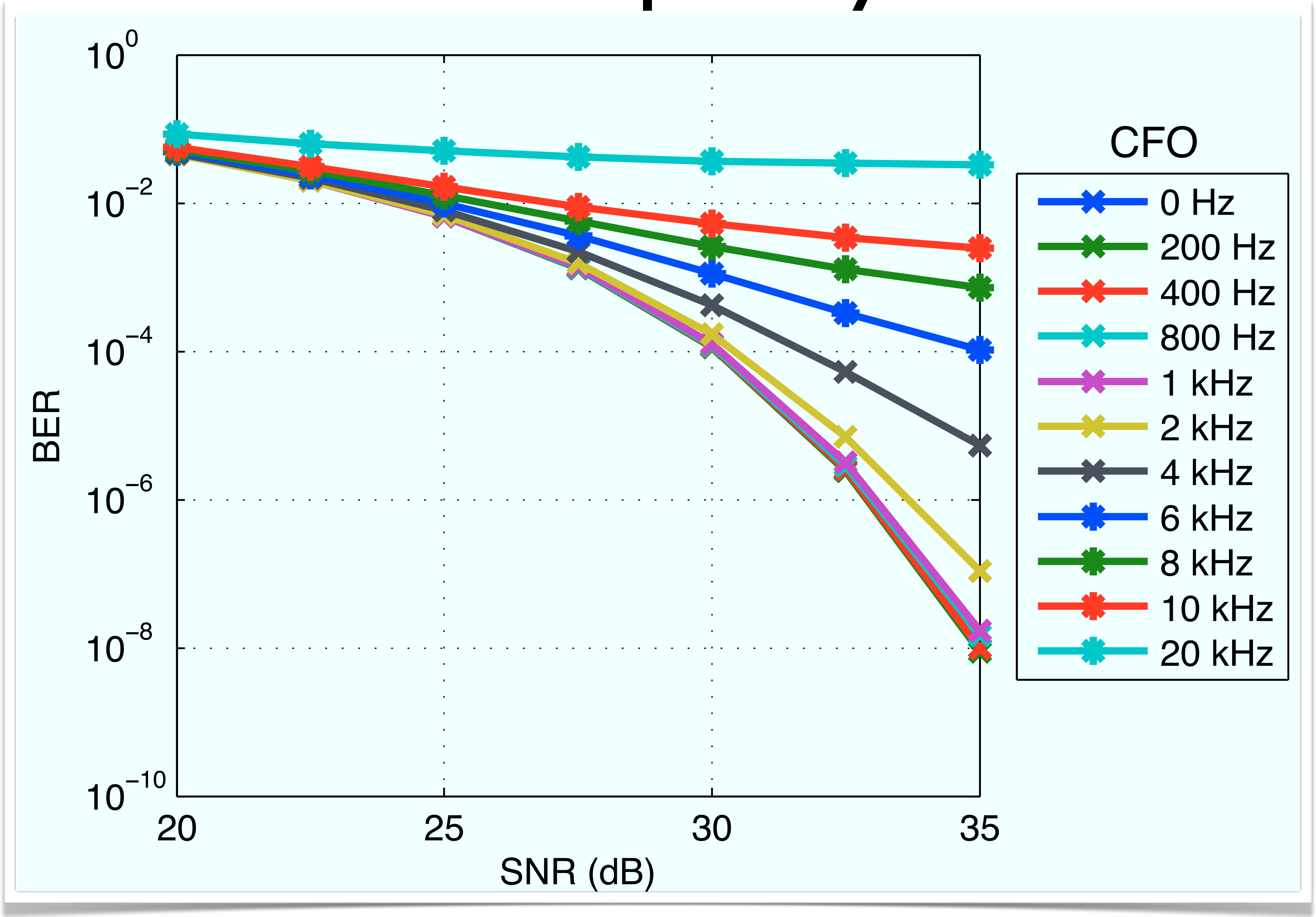
CFO with Cooperation

CFO with AF

CFO with DF

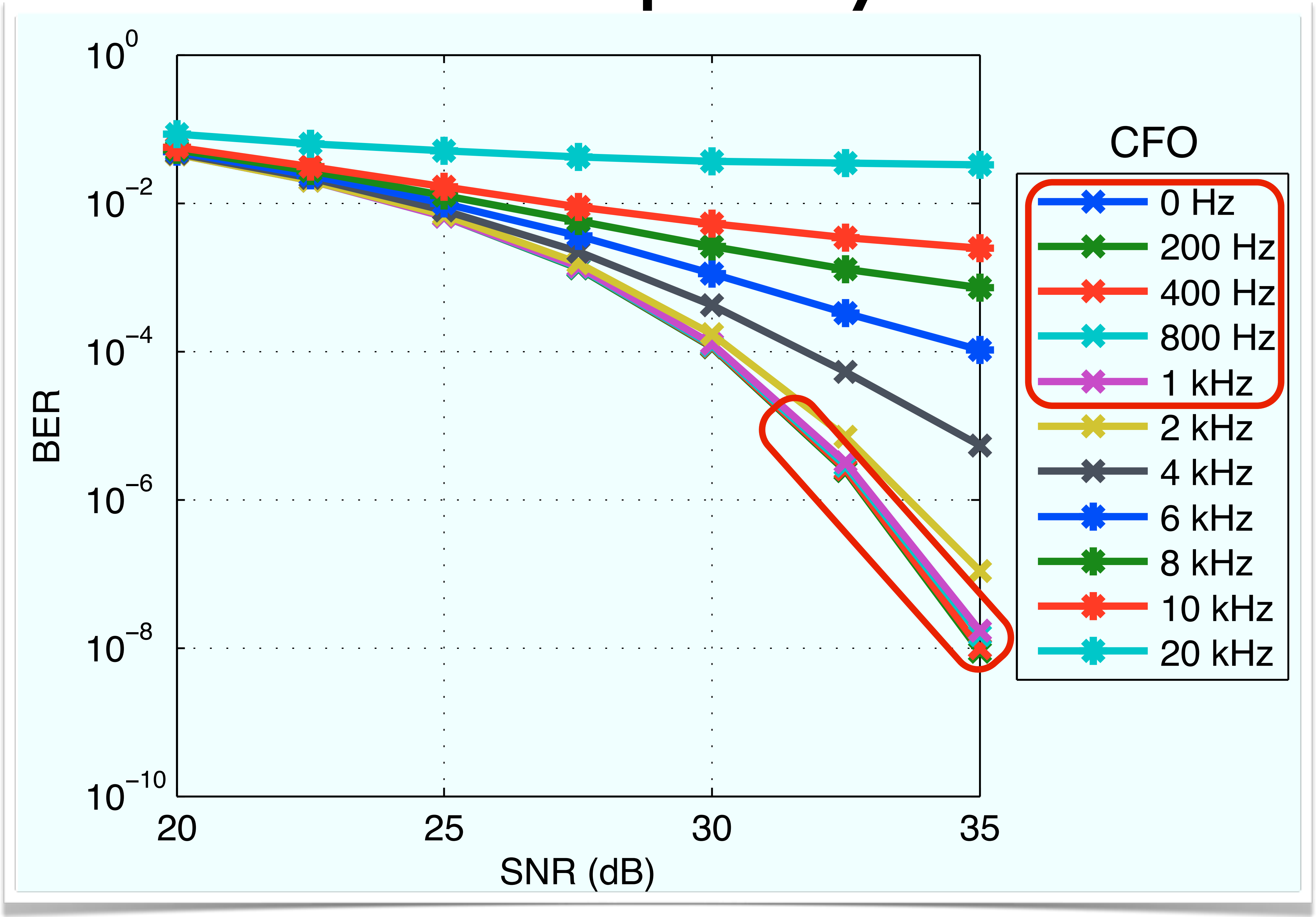
- ICI
- Phase errors

# Carrier Frequency Offset



- Monte Carlo sim of SISO OFDM
- AWGN
- 1500 byte 16-QAM packets
- Perfect phase error correction

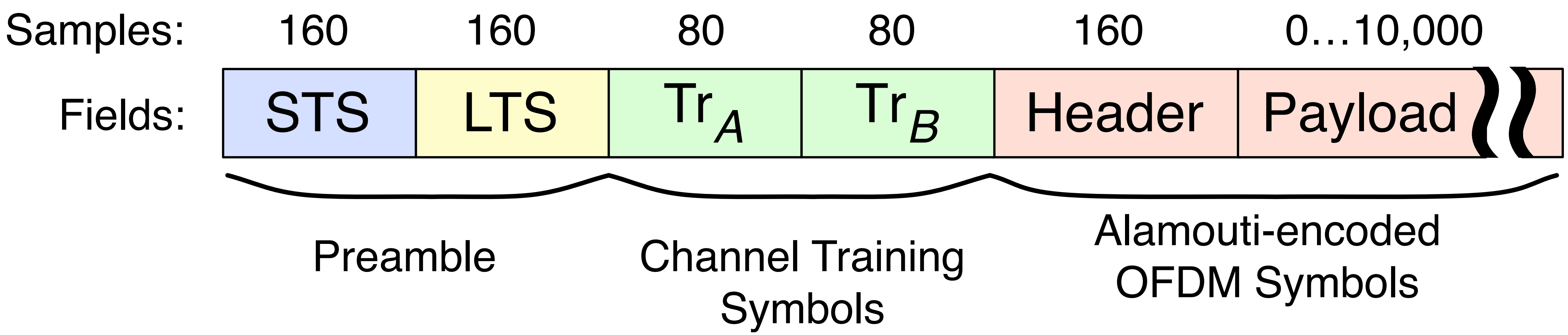
# Carrier Frequency Offset



- Monte Carlo sim of SISO OFDM
- AWGN
- 1500 byte 16-QAM packets
- Perfect phase error correction

# Carrier Frequency Offset

## *Time Domain CFO Estimation*



| Samples: | 160 | 160 | 80 | 80 | 160 | 0…10,000 |
|---|---|---|---|---|---|---|
| Fields: | STS | LTS | $\mathrm{Tr}_A$ | $\mathrm{Tr}_B$ | Header | Payload |

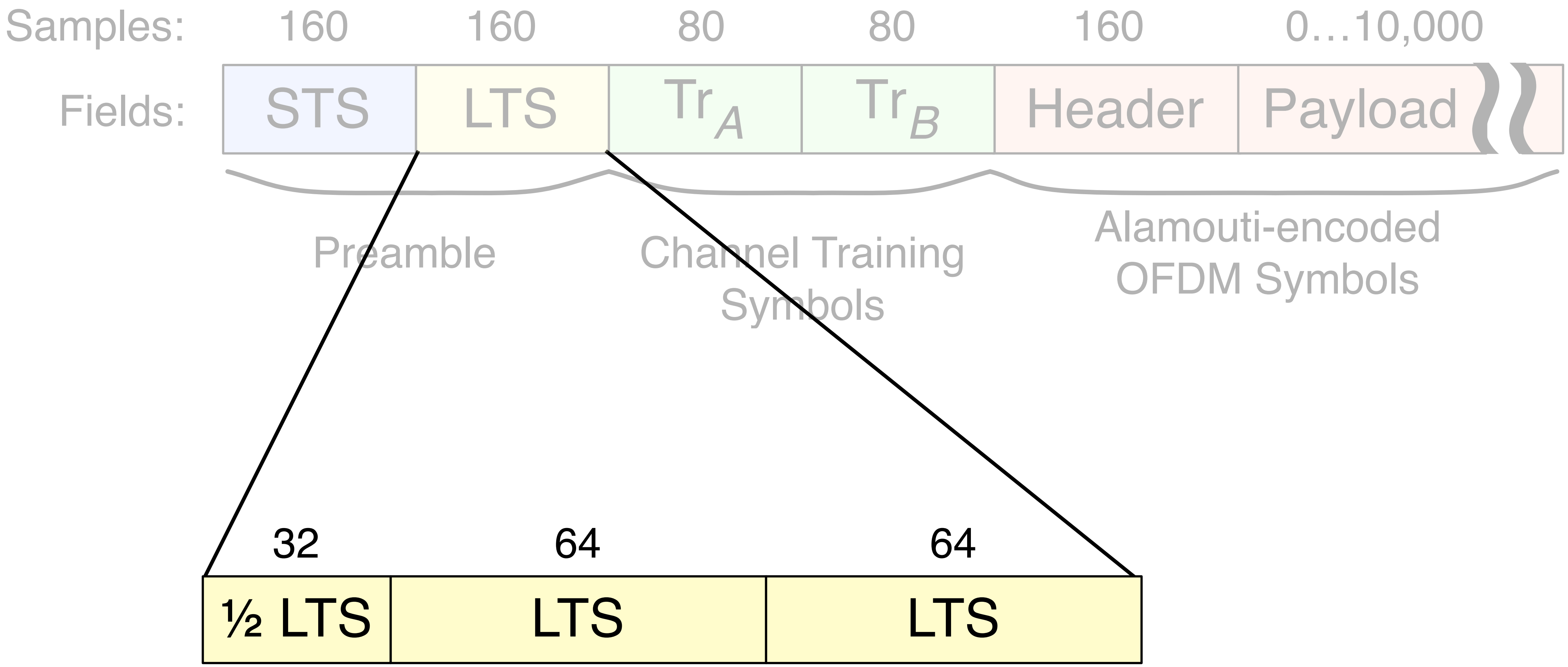
Preamble

Channel Training Symbols

Alamouti-encoded OFDM Symbols

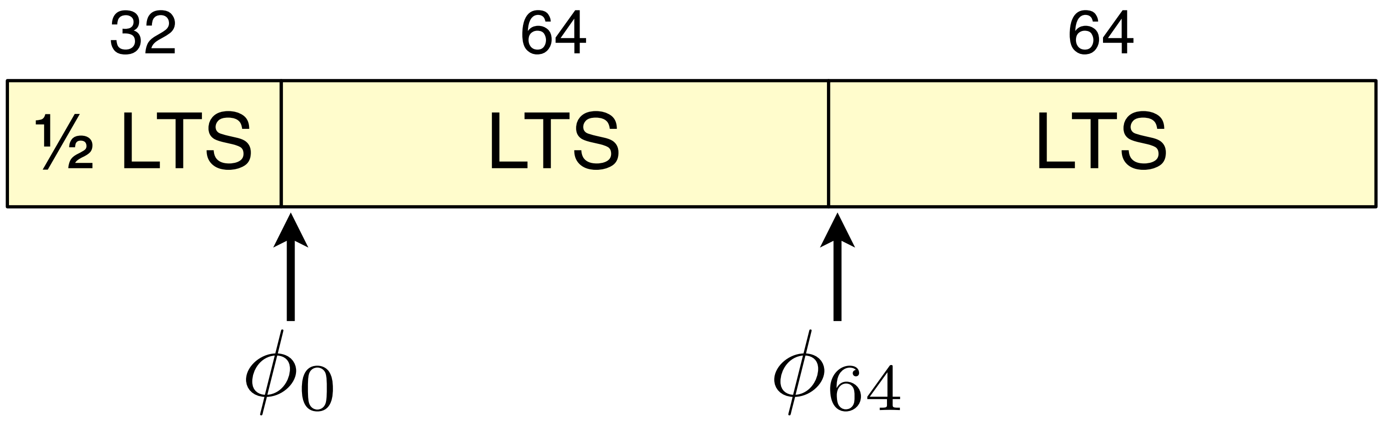
# Carrier Frequency Offset

*Time Domain CFO Estimation*



# Carrier Frequency Offset

*Time Domain CFO Estimation*

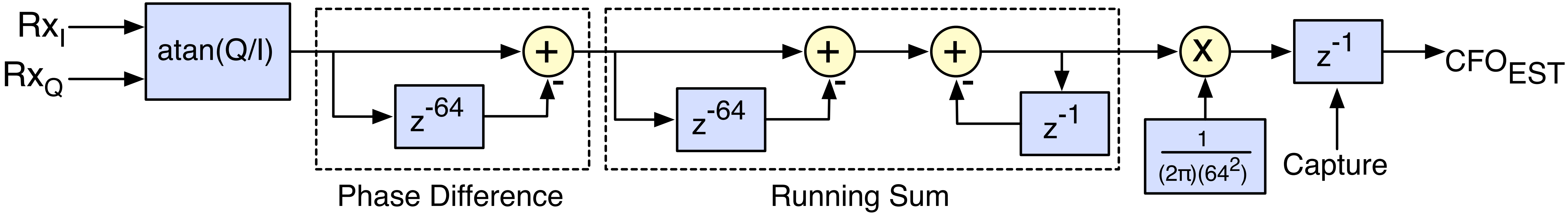


$$CFO \sim (\phi_{64} - \phi_0)$$

$$CFO_{EST} = \frac{f_s}{2\pi \cdot 64^2} \sum_{n=64}^{127} \phi_n - \phi_{(n-64)}$$

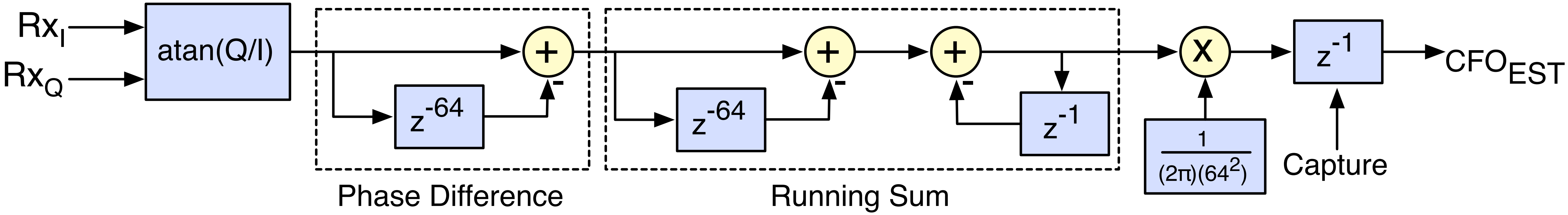
# Carrier Frequency Offset

*Time Domain CFO Estimation*



# Carrier Frequency Offset

*Time Domain CFO Estimation*

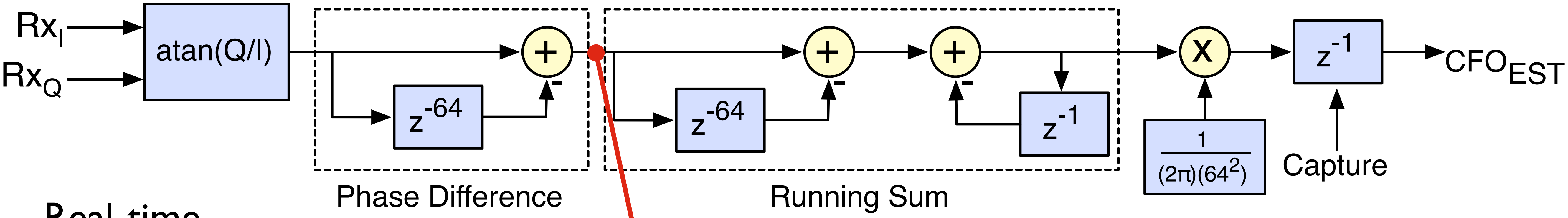


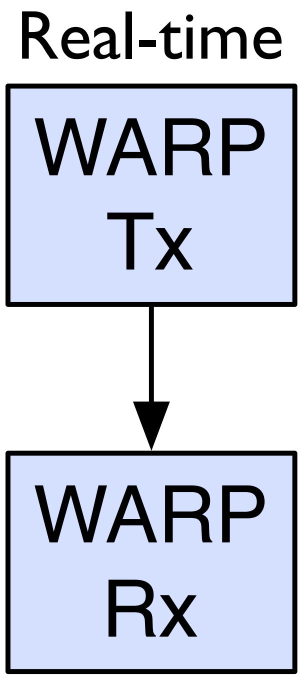
Something weird happened in characterizing the estimator...

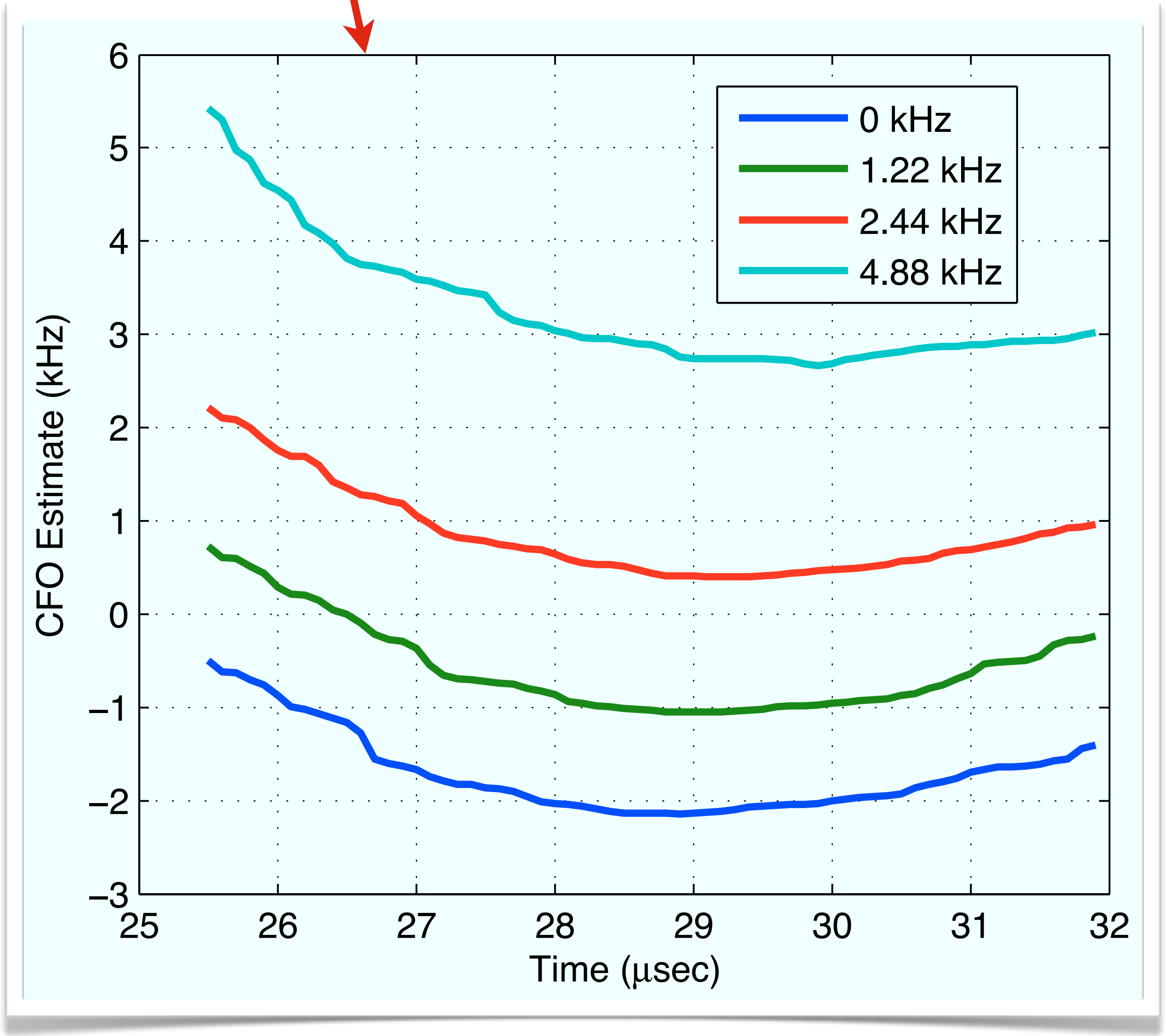
It didn’t work...at all.

# Carrier Frequency Offset

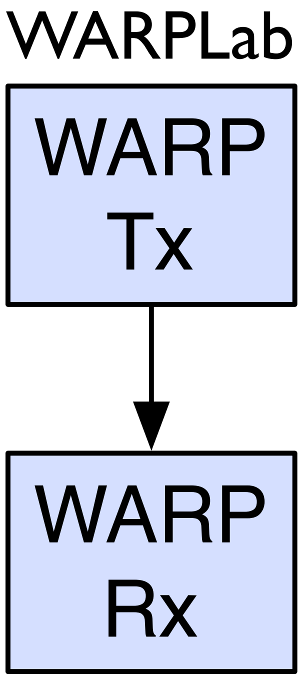
*Time Domain CFO Estimation*





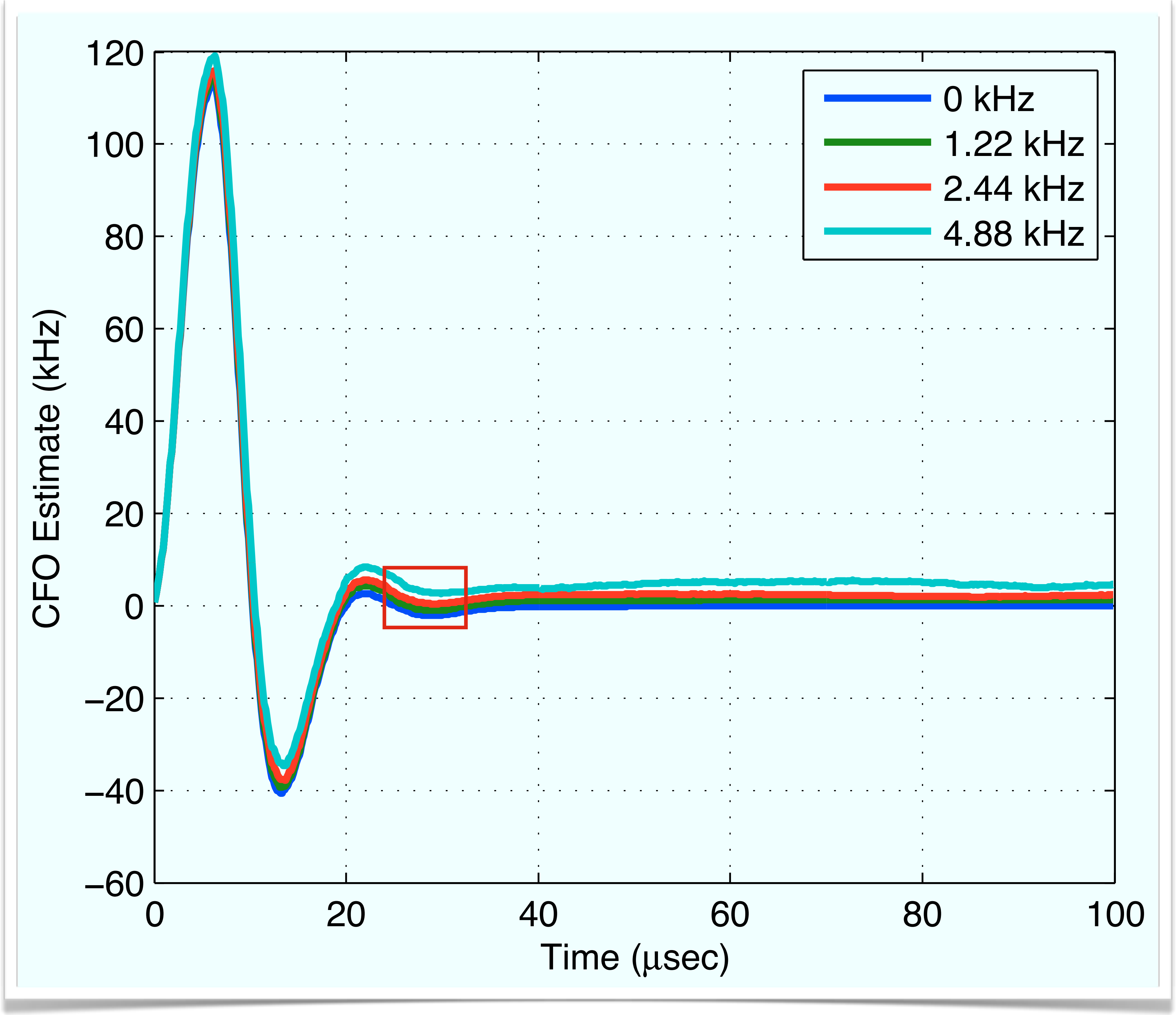


# Carrier Frequency Offset


WARPLab

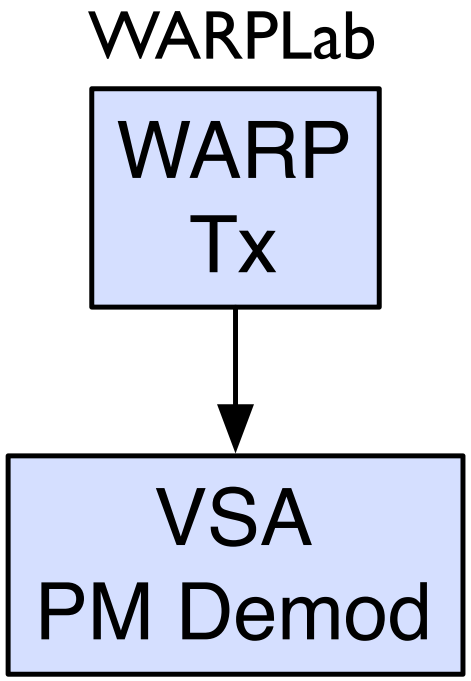
WARP Tx

WARP Rx


*Radio Transients*

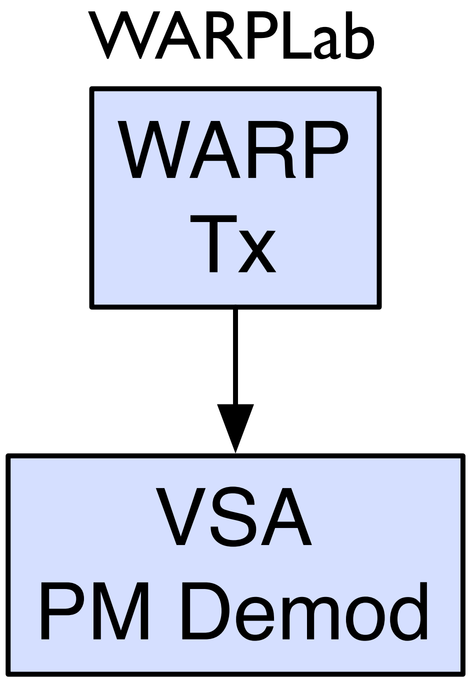
# Carrier Frequency Offset

*Radio Transients*



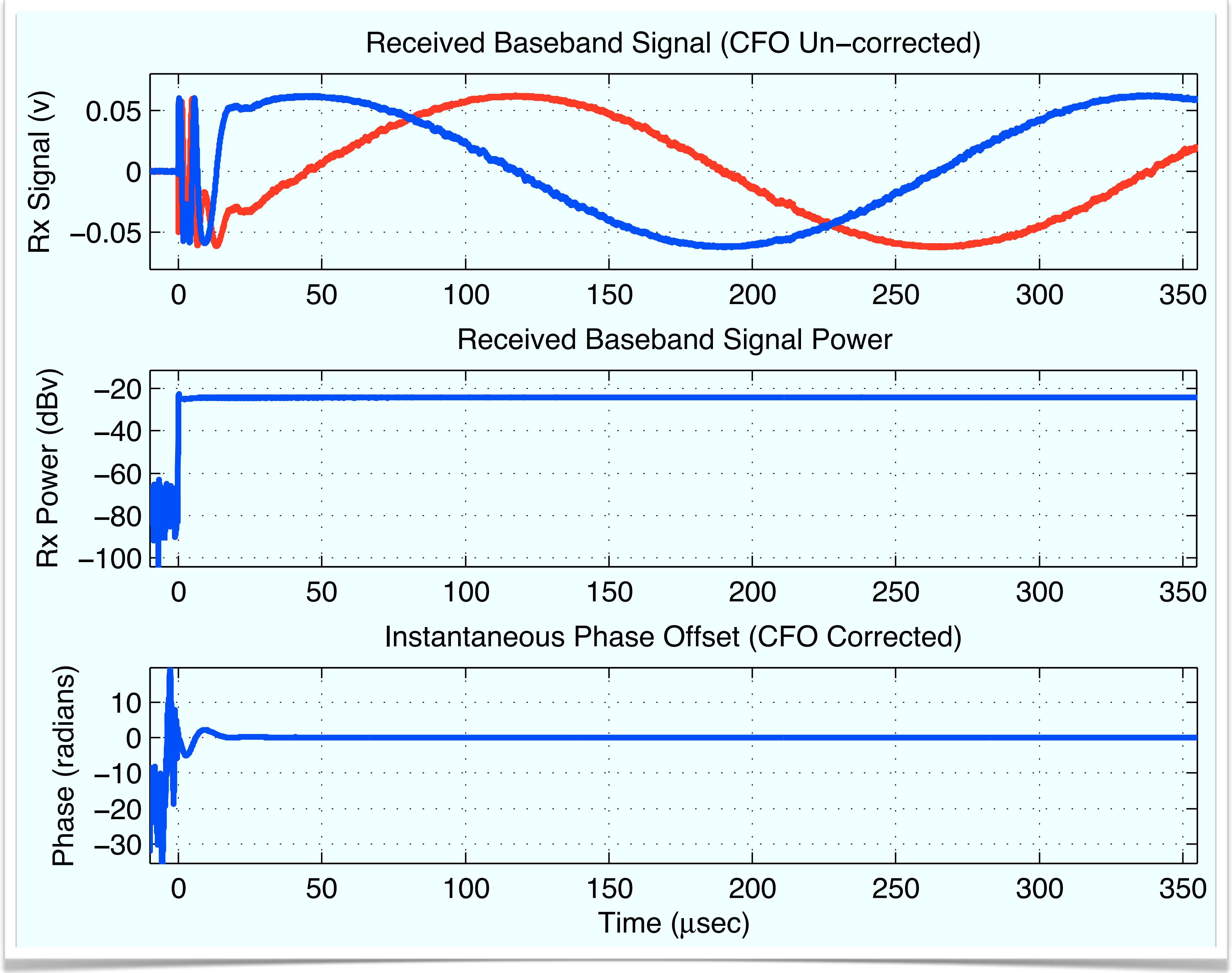
# Carrier Frequency Offset

*Radio Transients*


WARPLab

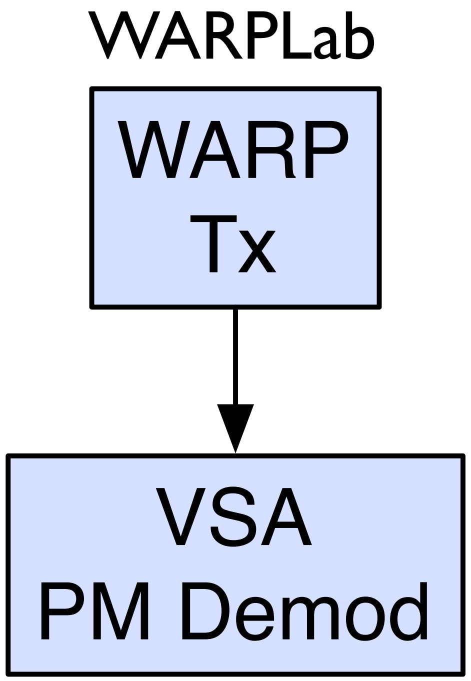
WARP
Tx

VSA
PM Demod



# Carrier Frequency Offset

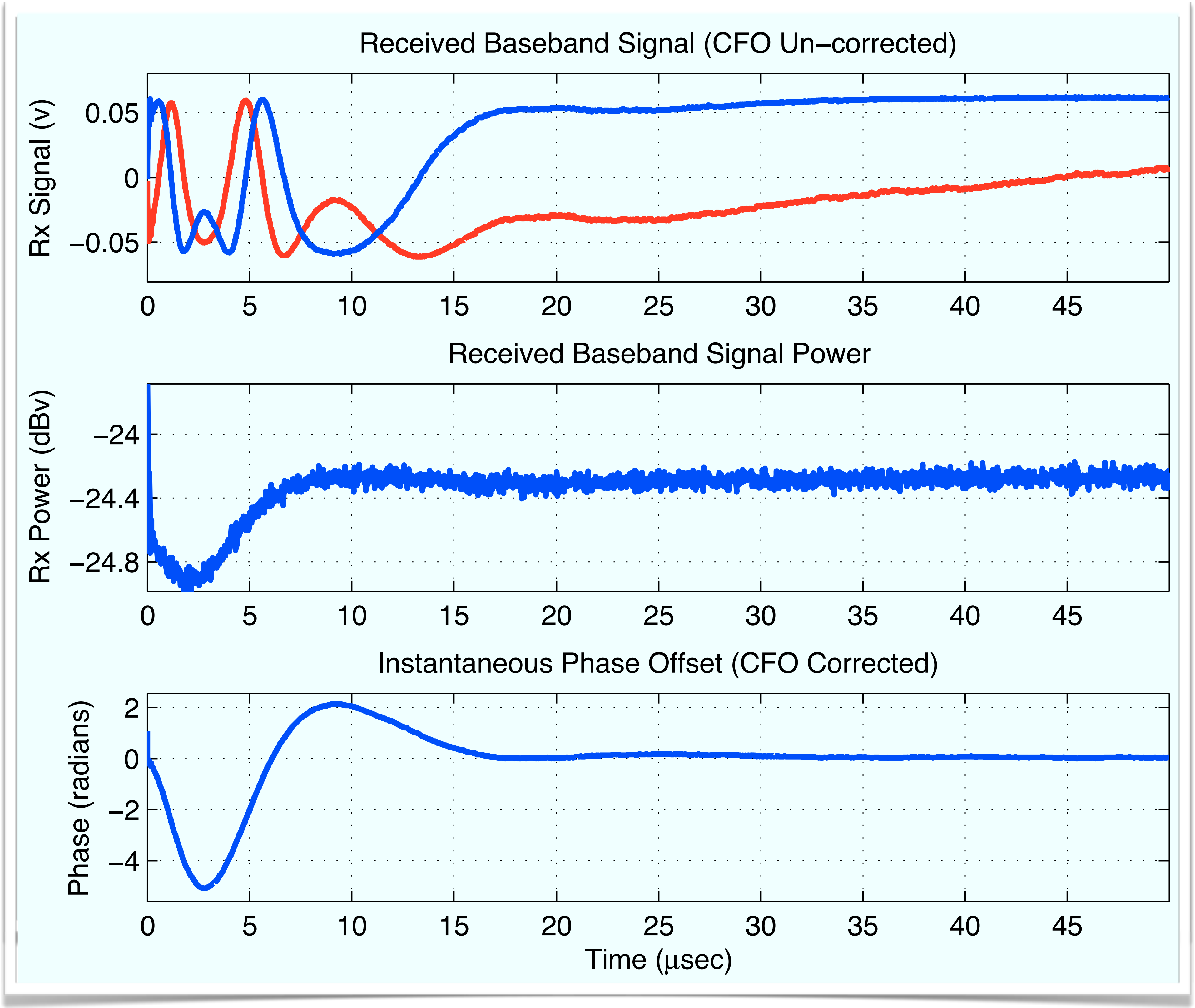
*Radio Transients*

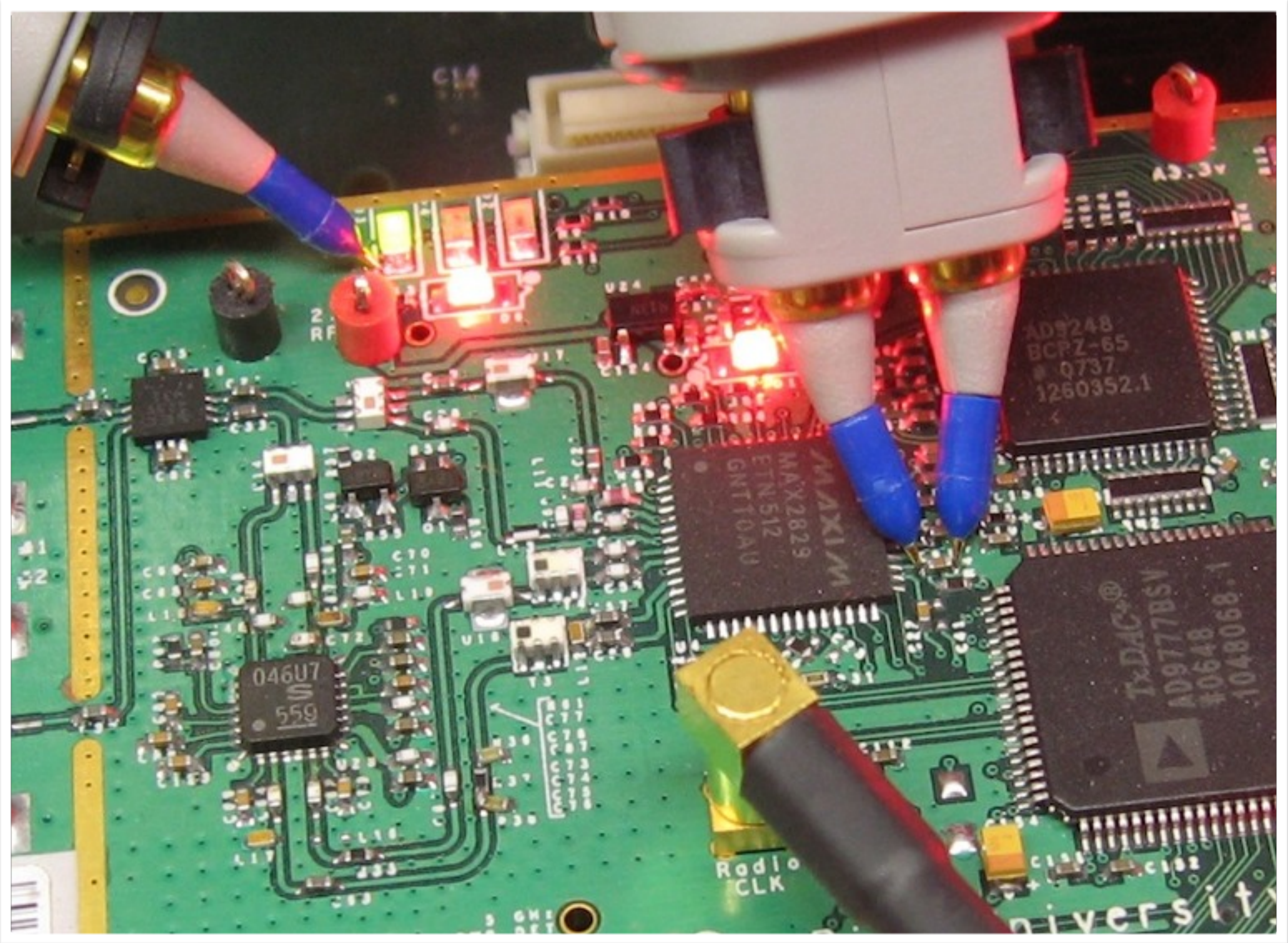


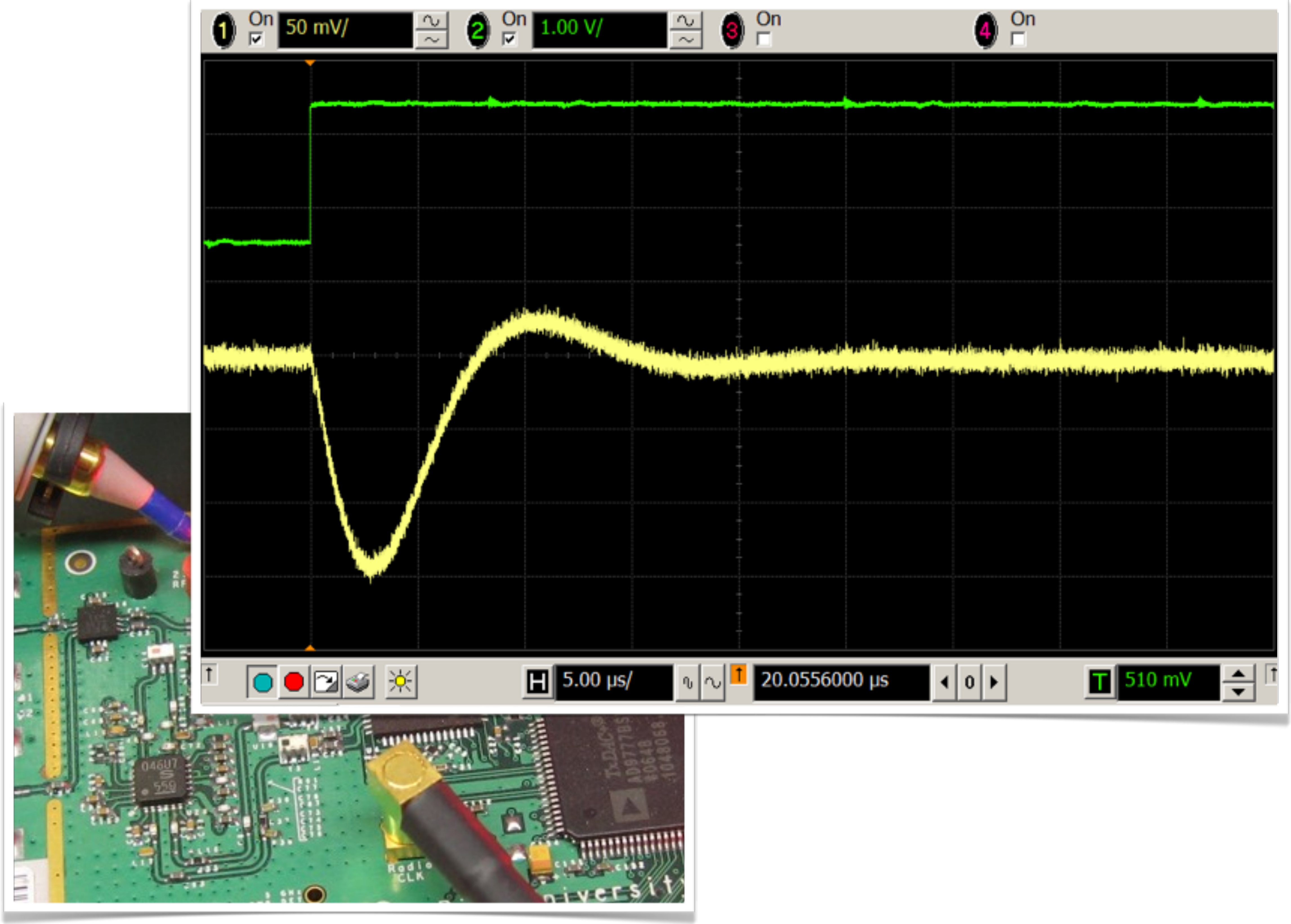
WARPLab

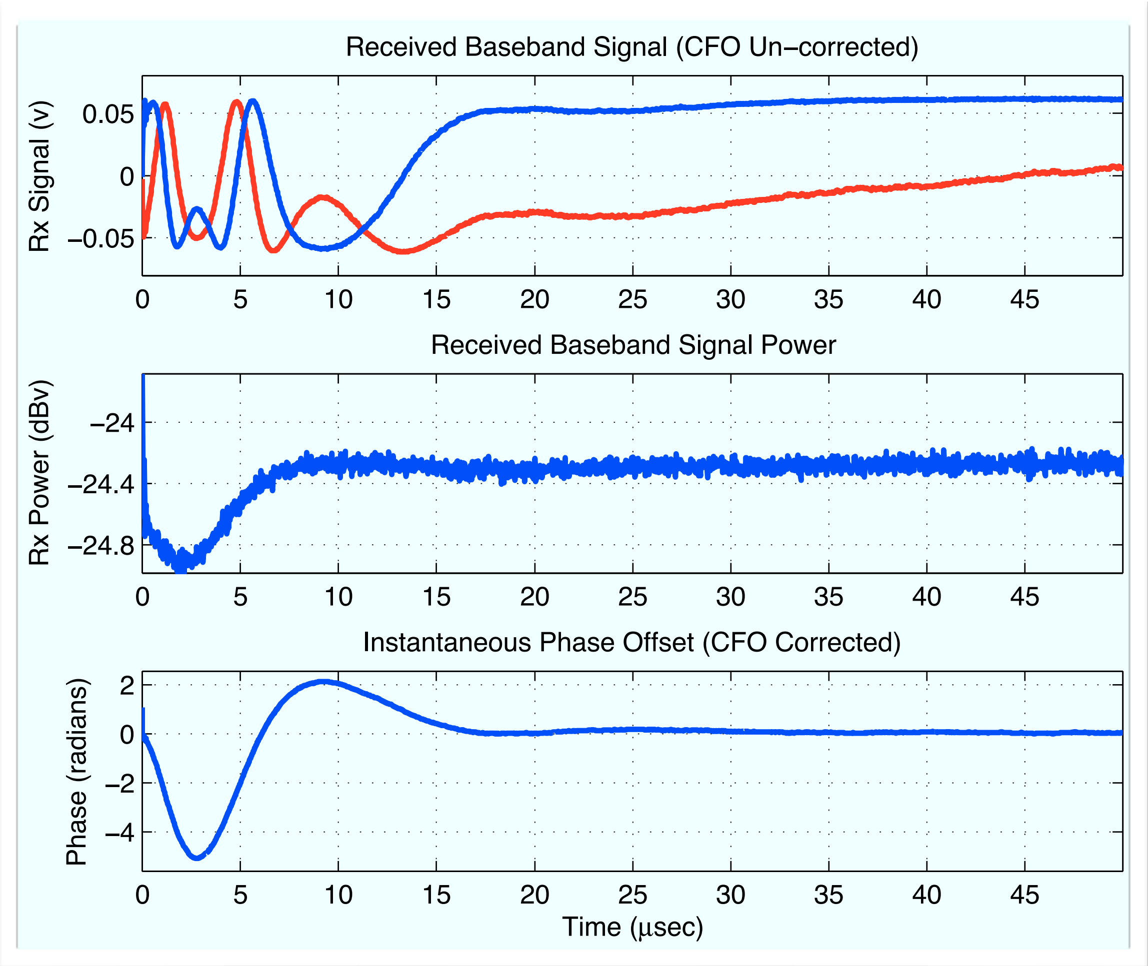
WARP Tx

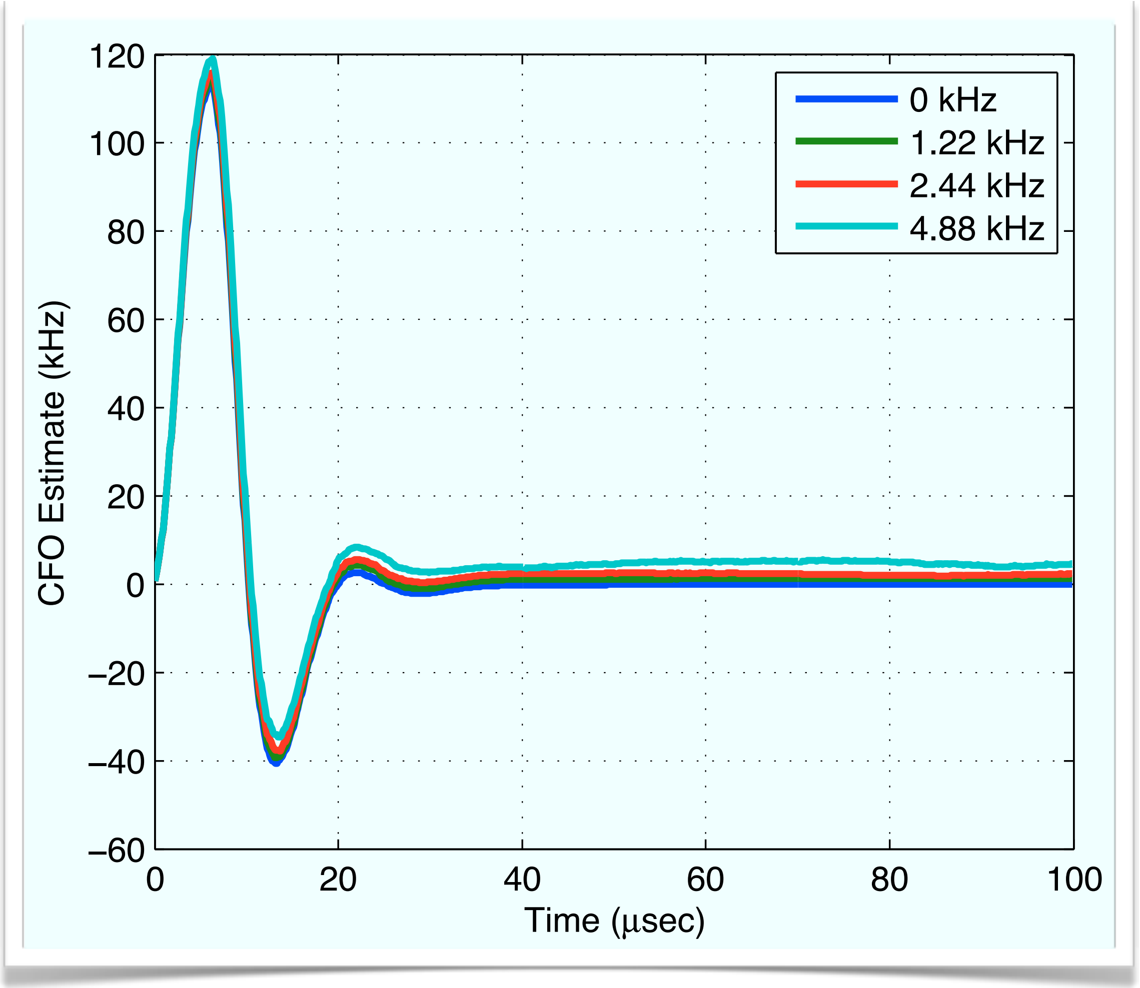
VSA PM Demod

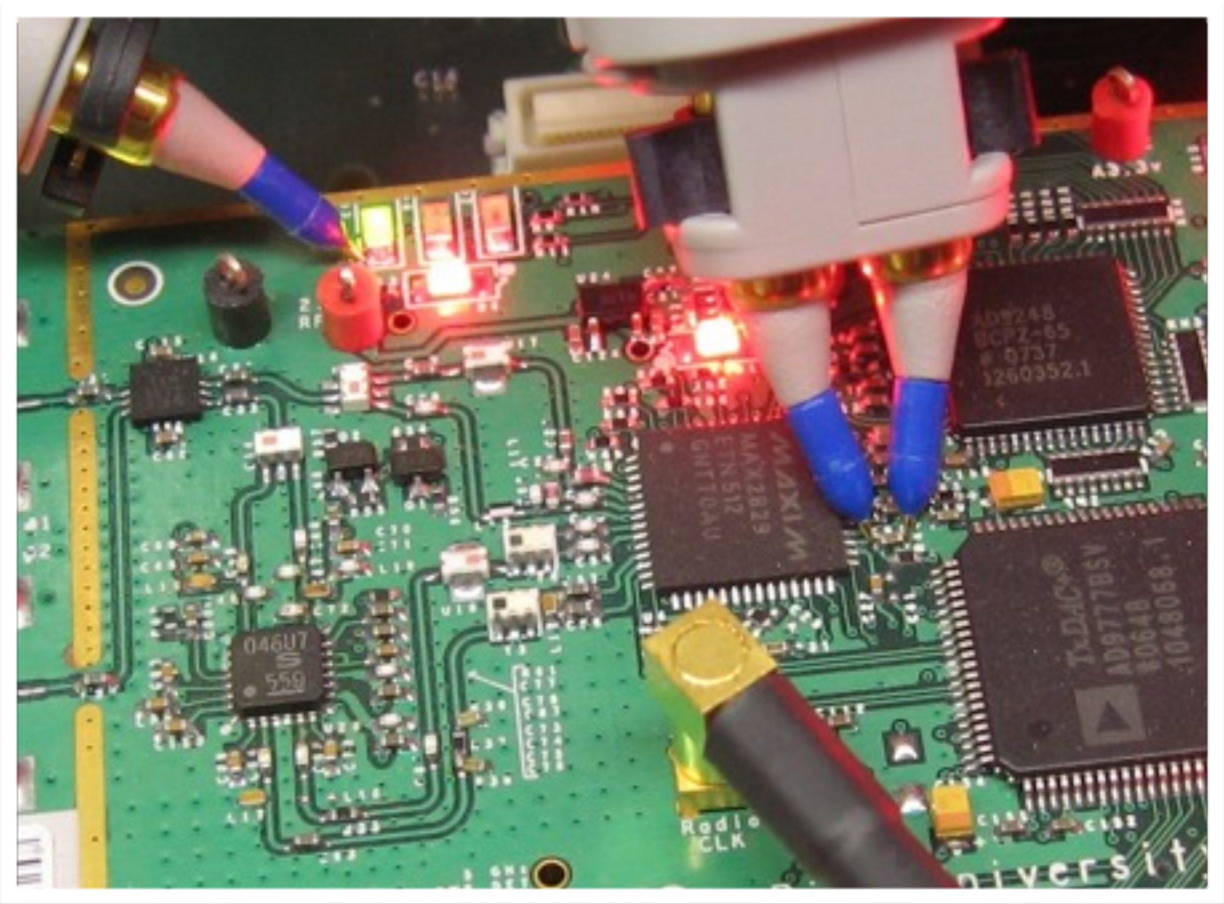


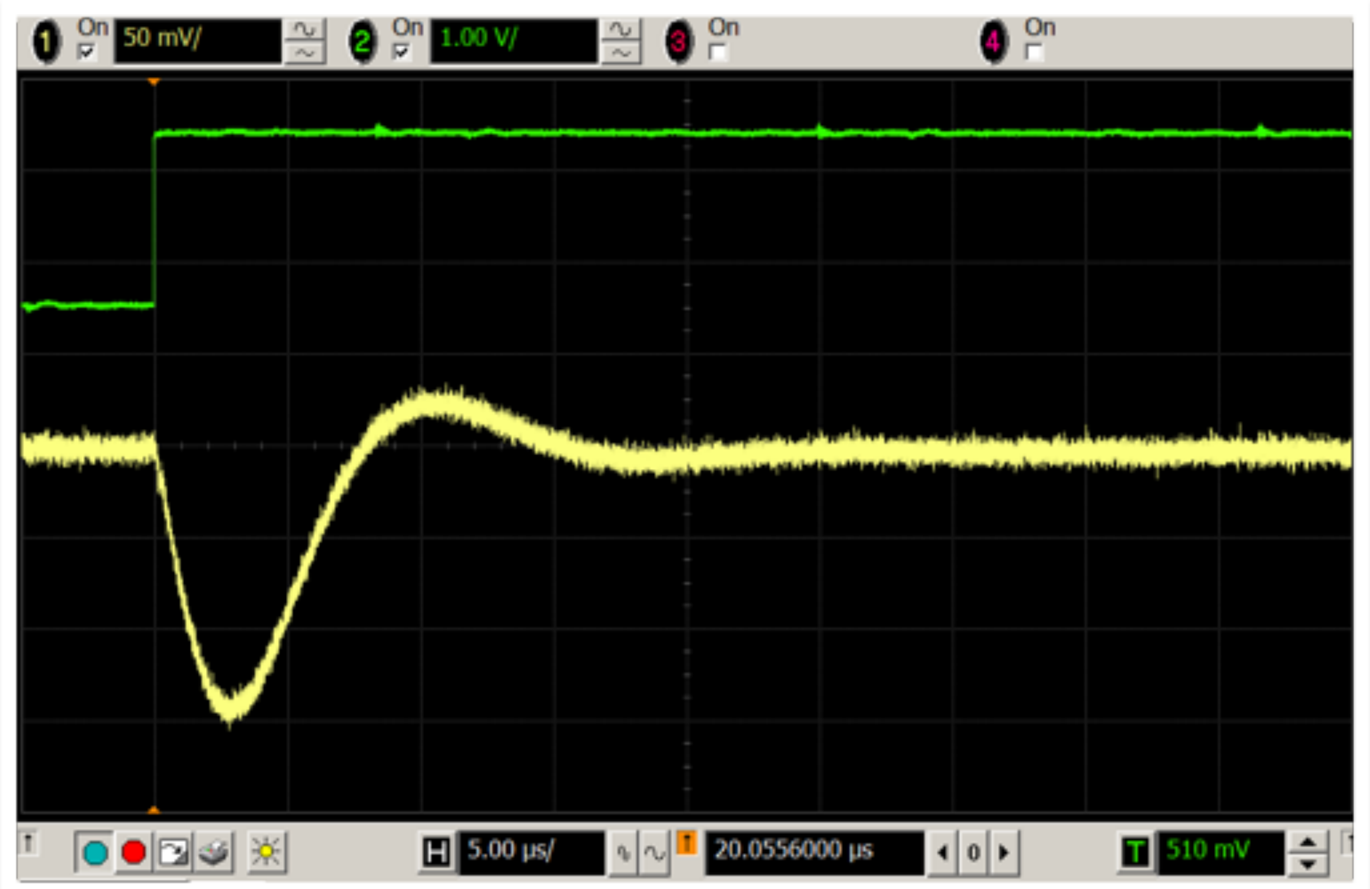

MAXIM
MAX2829
ETN512
GNT10AU
AD9248
BCPZ-65
0737
1260352.1
TxDAC+
AD9777BSV
#0648
1048068.1
046U7
S
559
A3.3v
Radio
CLK
niversity


1 On ☑ 50 mV/
2 On ☑ 1.00 V/
3 On ☐
4 On ☐
H 5.00 µs/
20.0556000 µs
0
T 510 mV
Radio CLK
niversit





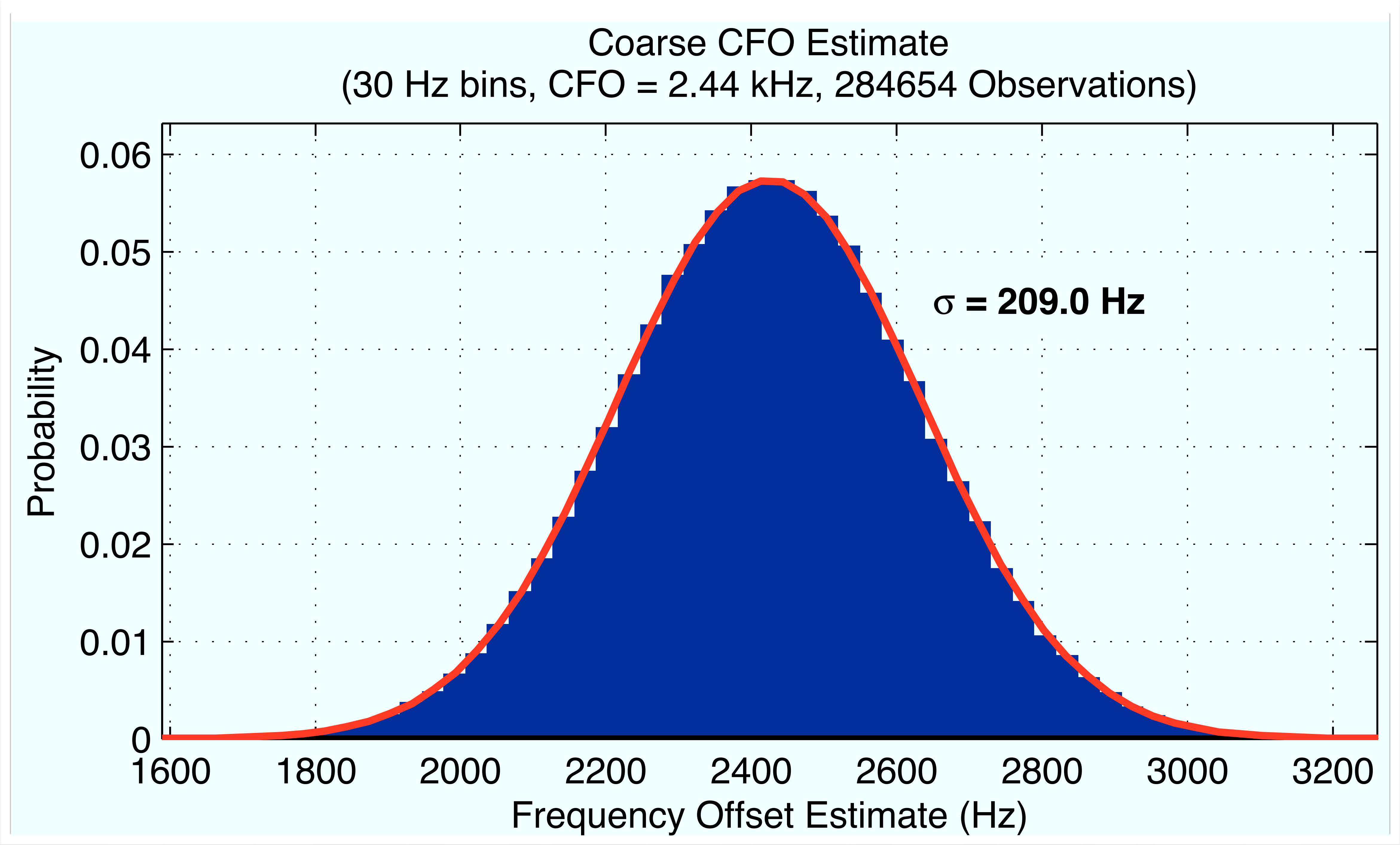




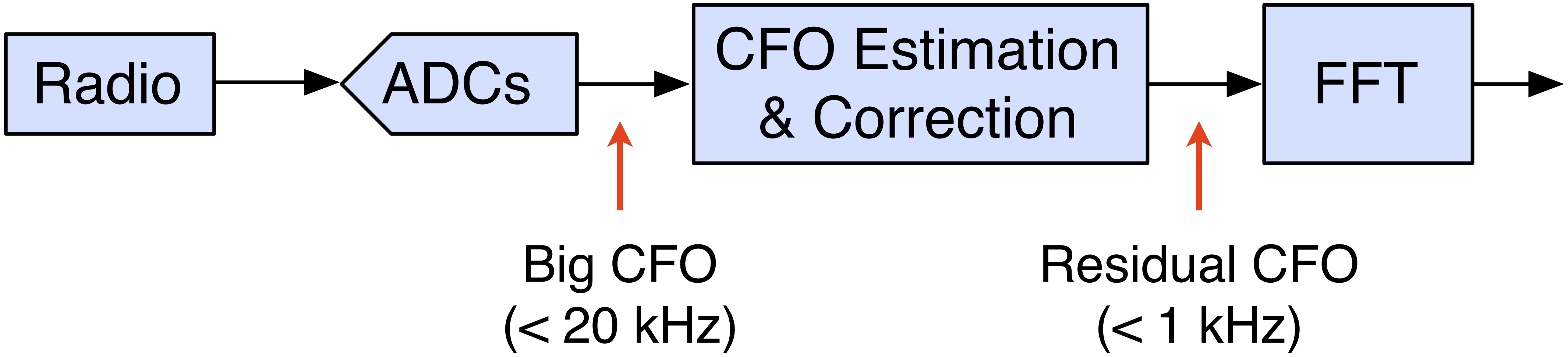
1 On ☑ 50 mV/
2 On ☑ 1.00 V/
3 On ☐
4 On ☐
H 5.00 µs/
20.0556000 µs
0
T 510 mV

# Carrier Frequency Offset

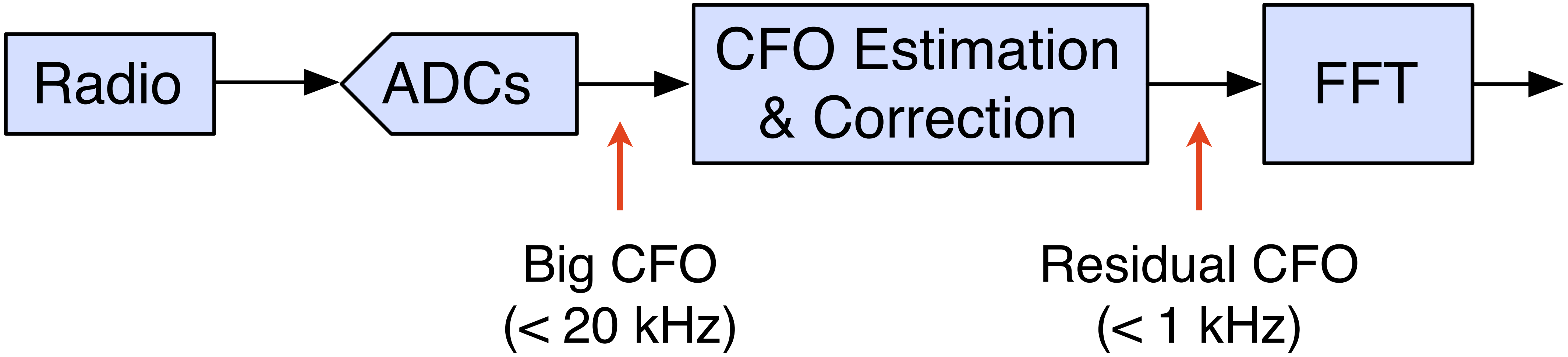
*Time Domain CFO Estimation*



# Carrier Frequency Offset

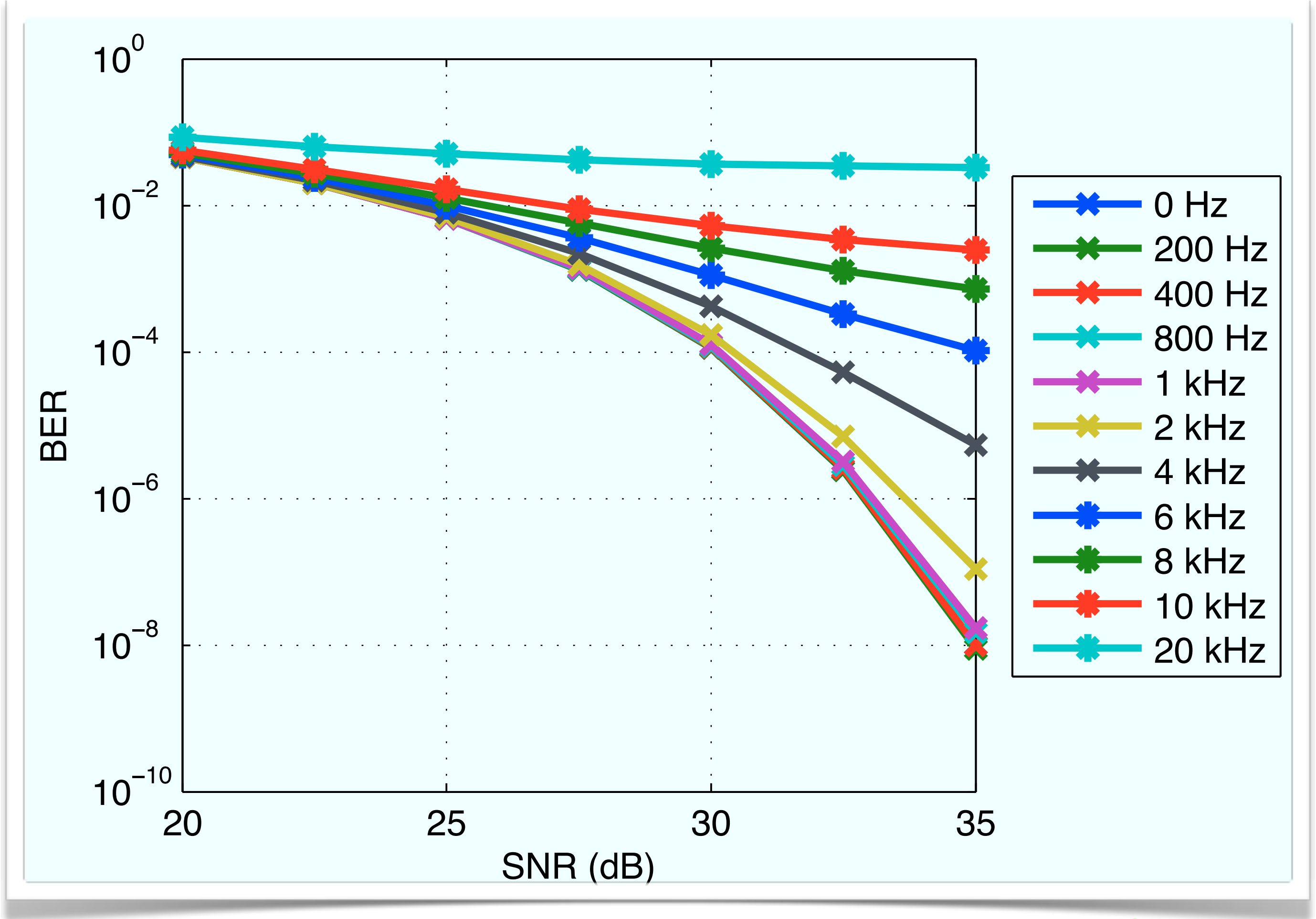


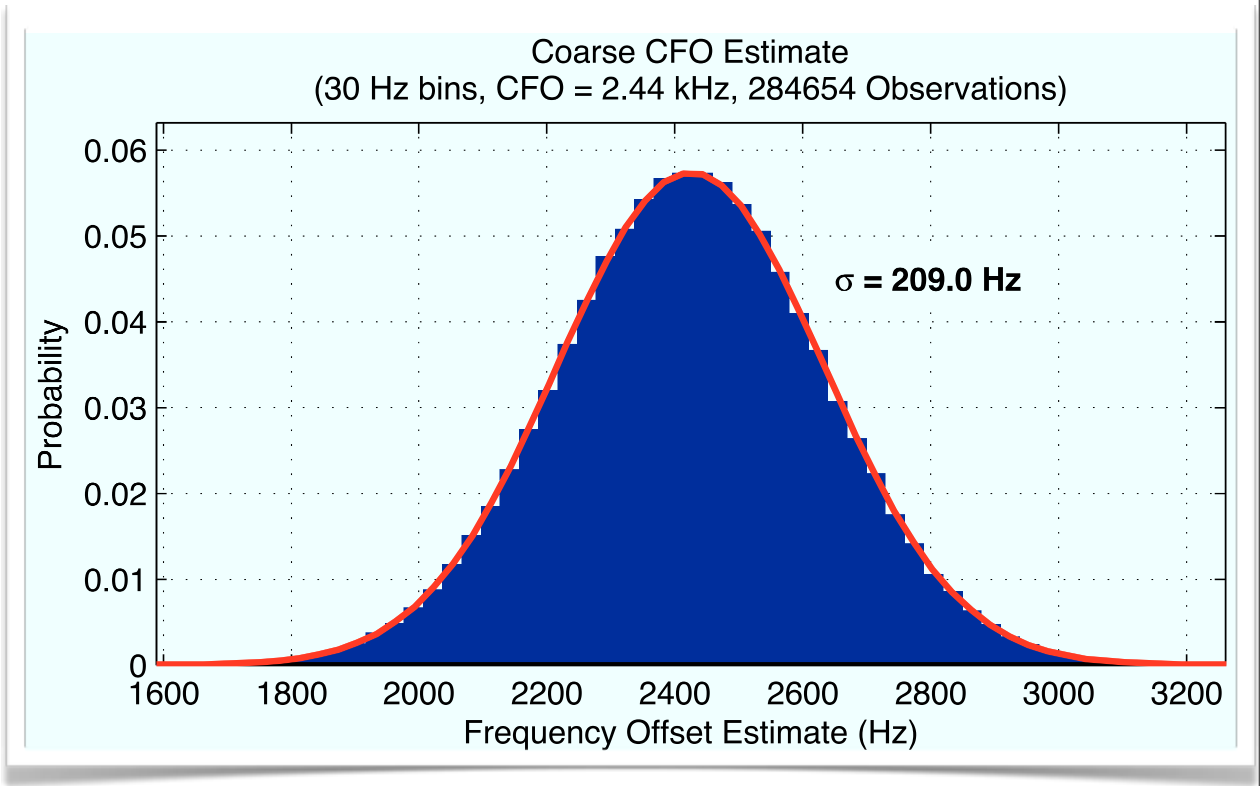
# Carrier Frequency Offset



Big CFO
(< 20 kHz)

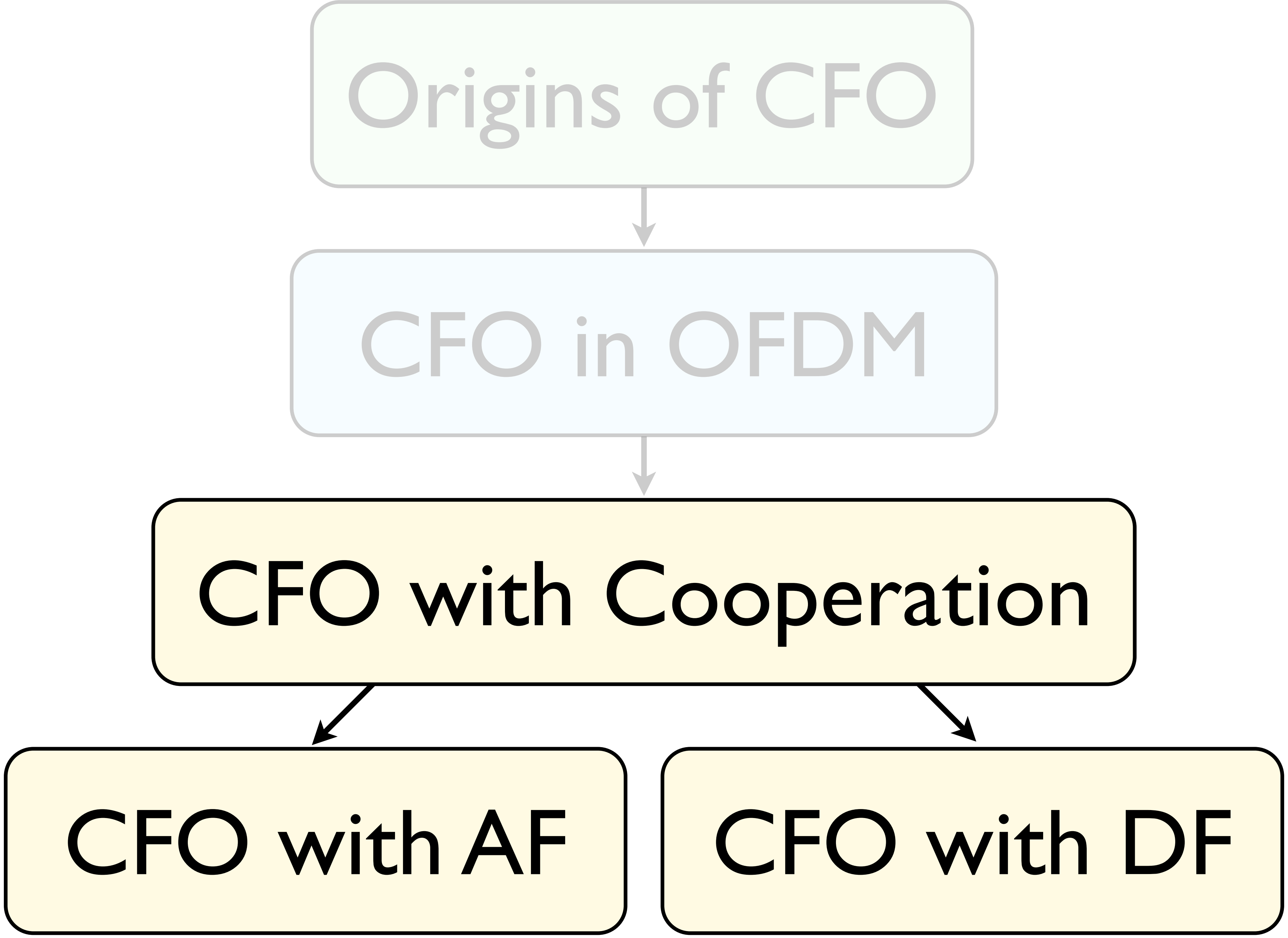
Residual CFO
(< 1 kHz)




Coarse CFO Estimate
(30 Hz bins, CFO = 2.44 kHz, 284654 Observations)

✓ Success!

# Carrier Frequency Offset



Origins of CFO

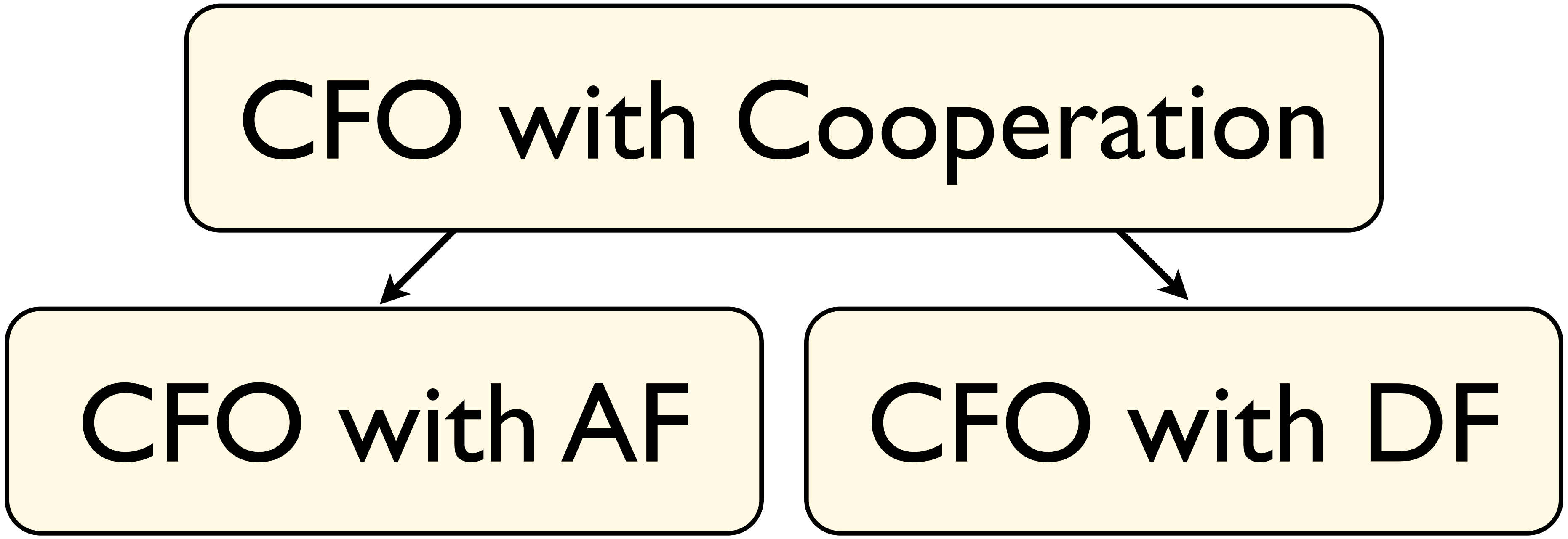
CFO in OFDM

CFO with Cooperation

CFO with AF

CFO with DF

# Carrier Frequency Offset



CFO with Cooperation

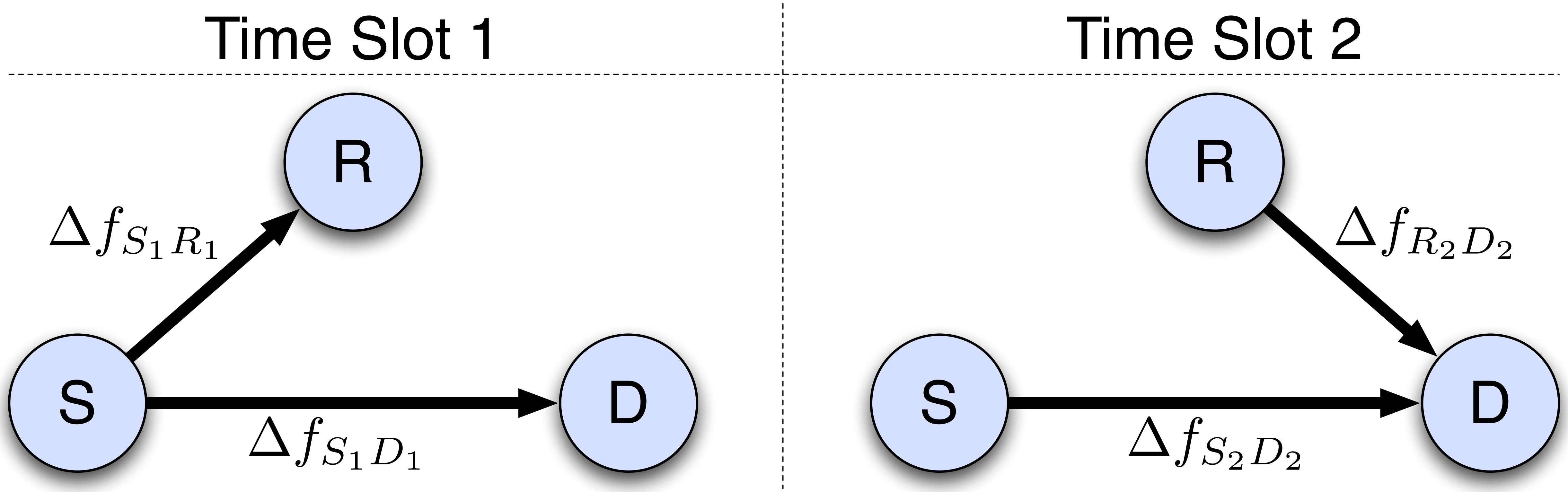
CFO with AF

CFO with DF

- CFO stability vs. time
- Offsets between cooperating nodes
  - Correcting it for AF/DF

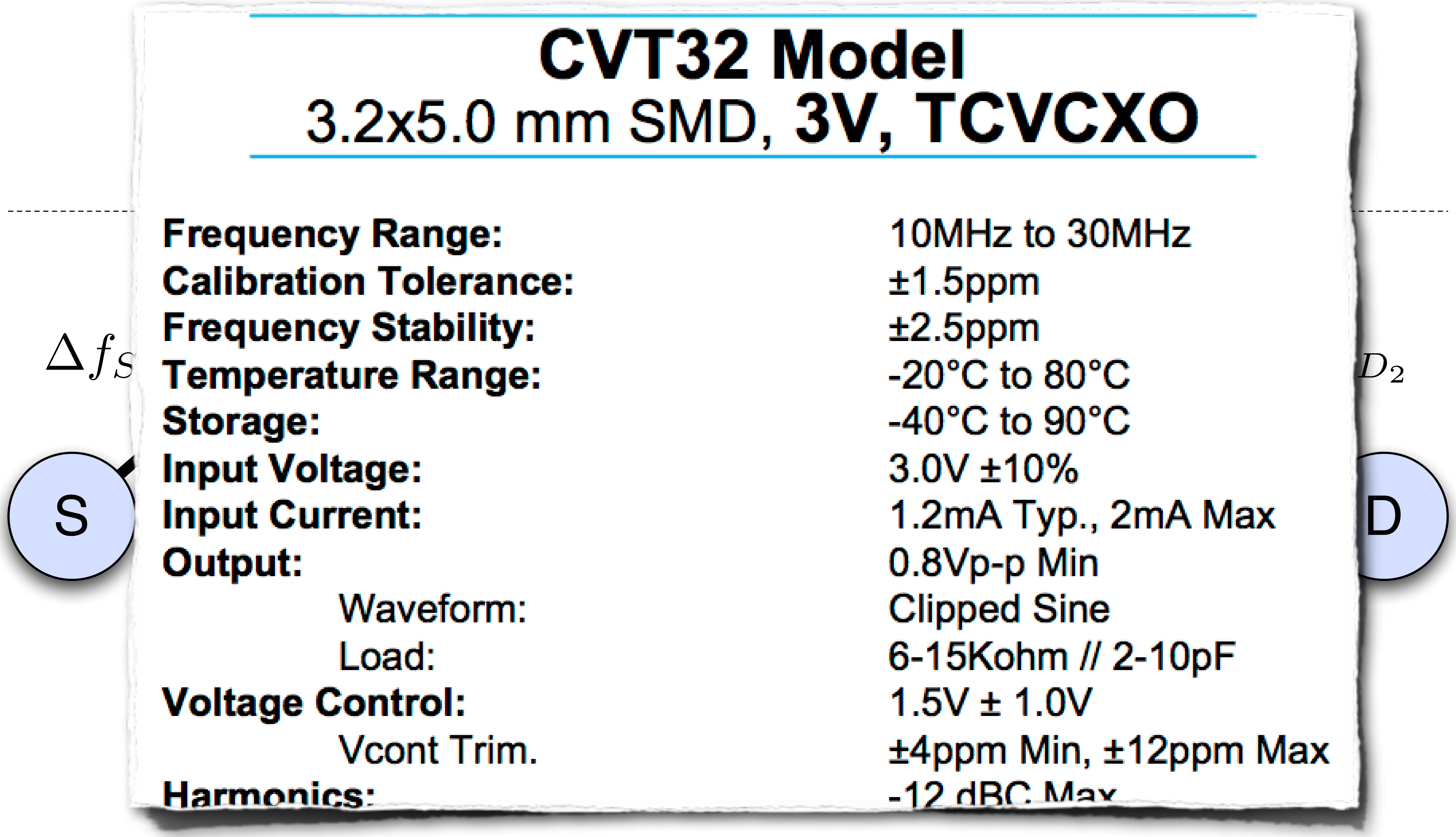
# Carrier Frequency Offset

*CFO in Cooperative Links*



# Carrier Frequency Offset

*CFO in Cooperative Links*



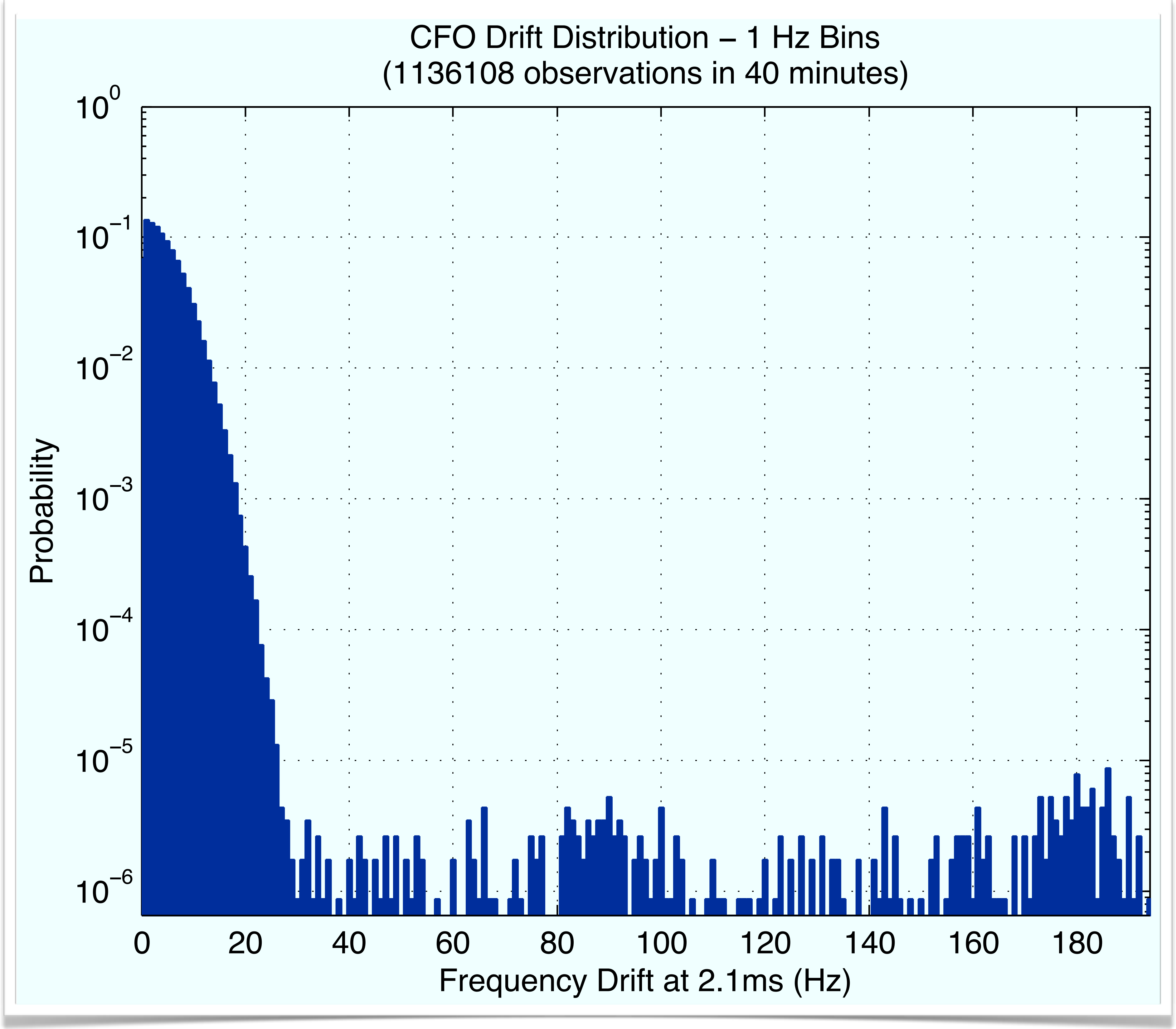
## CVT32 Model

3.2x5.0 mm SMD, **3V, TCVCXO**

| | |
|---|---|
| **Frequency Range:** | 10MHz to 30MHz |
| **Calibration Tolerance:** | ±1.5ppm |
| **Frequency Stability:** | ±2.5ppm |
| **Temperature Range:** | -20°C to 80°C |
| **Storage:** | -40°C to 90°C |
| **Input Voltage:** | 3.0V ±10% |
| **Input Current:** | 1.2mA Typ., 2mA Max |
| **Output:** | 0.8Vp-p Min |
| Waveform: | Clipped Sine |
| Load: | 6-15Kohm // 2-10pF |
| **Voltage Control:** | 1.5V ± 1.0V |
| Vcont Trim. | ±4ppm Min, ±12ppm Max |
| **Harmonics:** | -12 dBC Max |

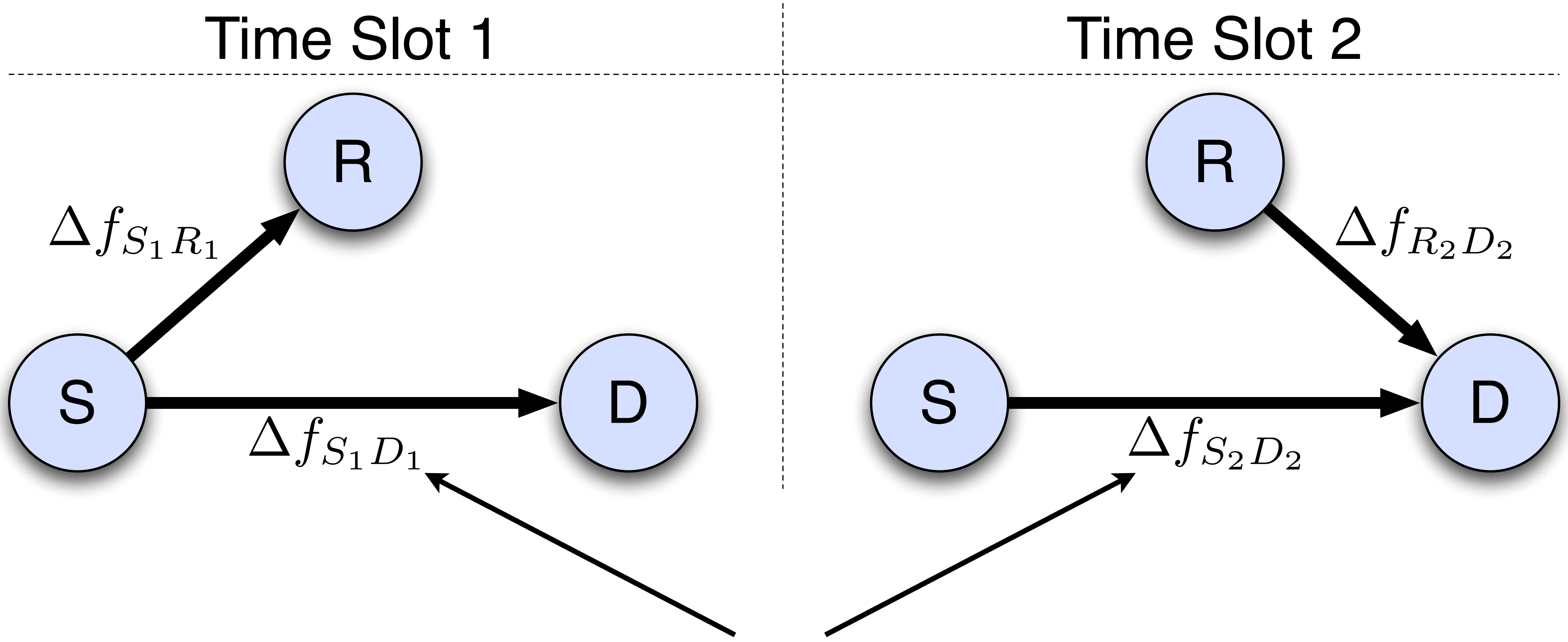
# Carrier Frequency Offset

*CFO in Cooperative Links*



# Carrier Frequency Offset

*CFO in Cooperative Links*



Same

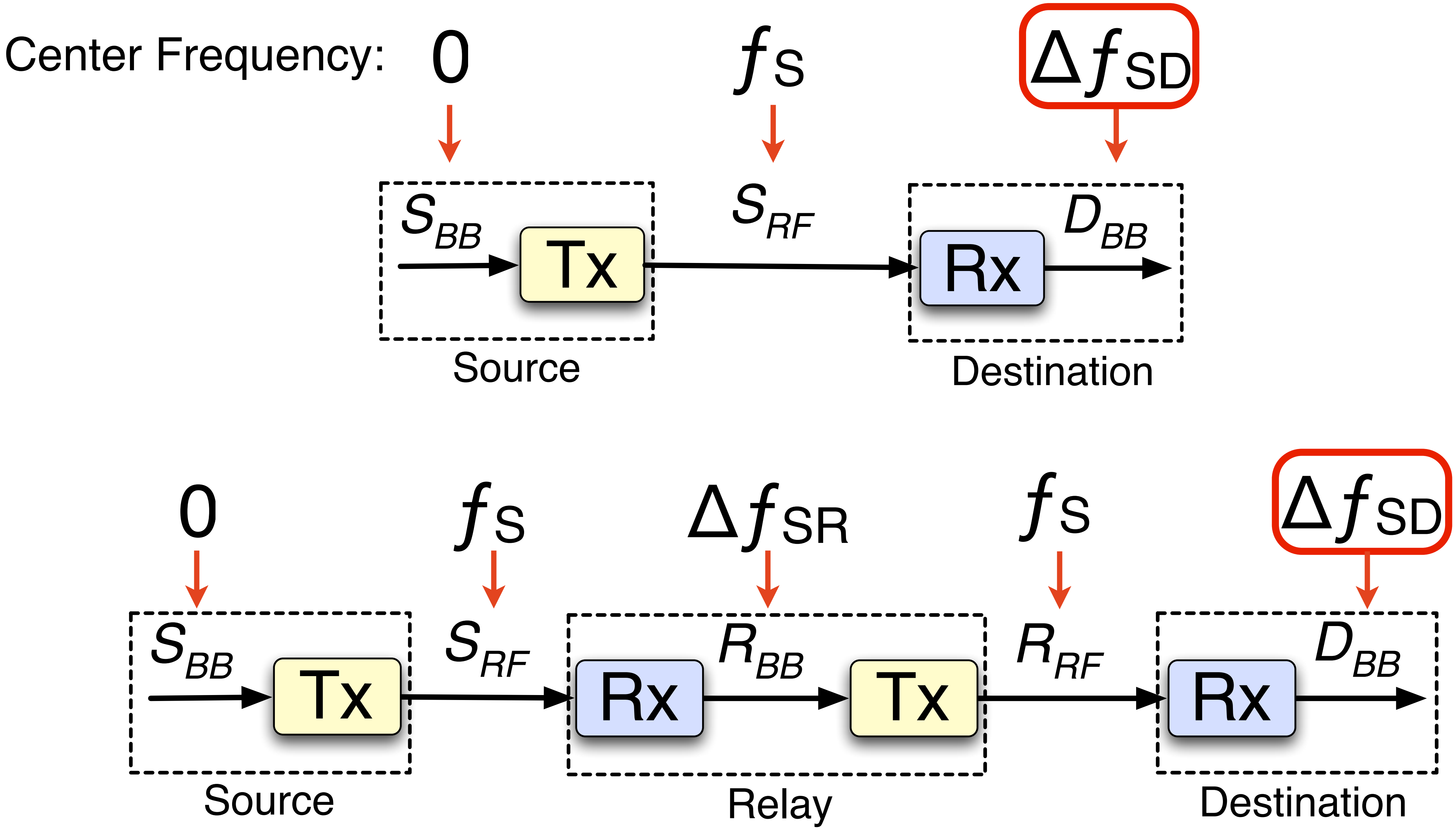
(with high probability for adjacent ~2 msec slots)

# Carrier Frequency Offset

- CFO with Amplify and Forward
- CFO with Decode and Forward

# Carrier Frequency Offset

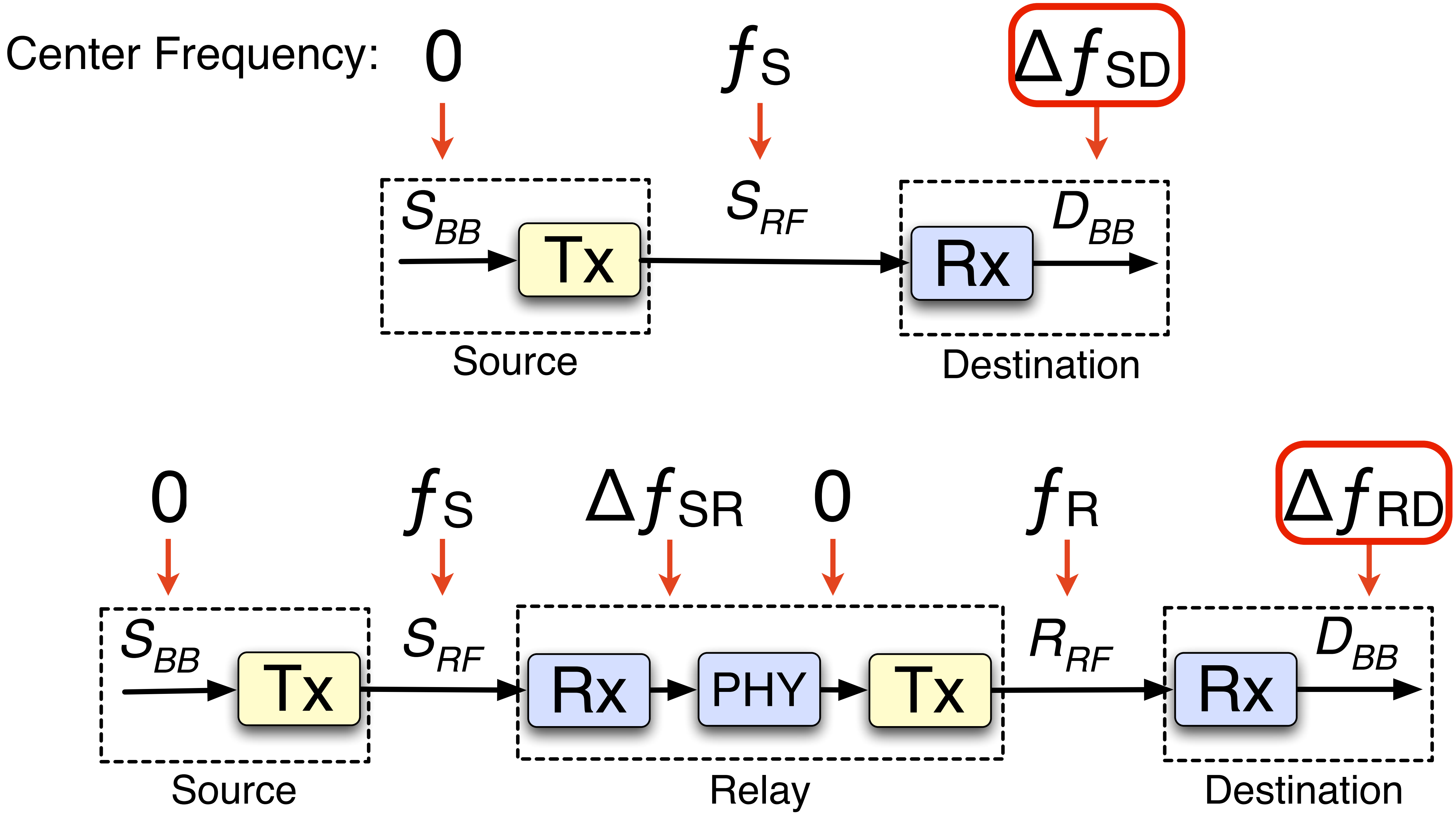
*CFO in Amplify and Forward*



- Inter-Tx CFO in AF is almost trivial
- Coherence time of oscillators still matters

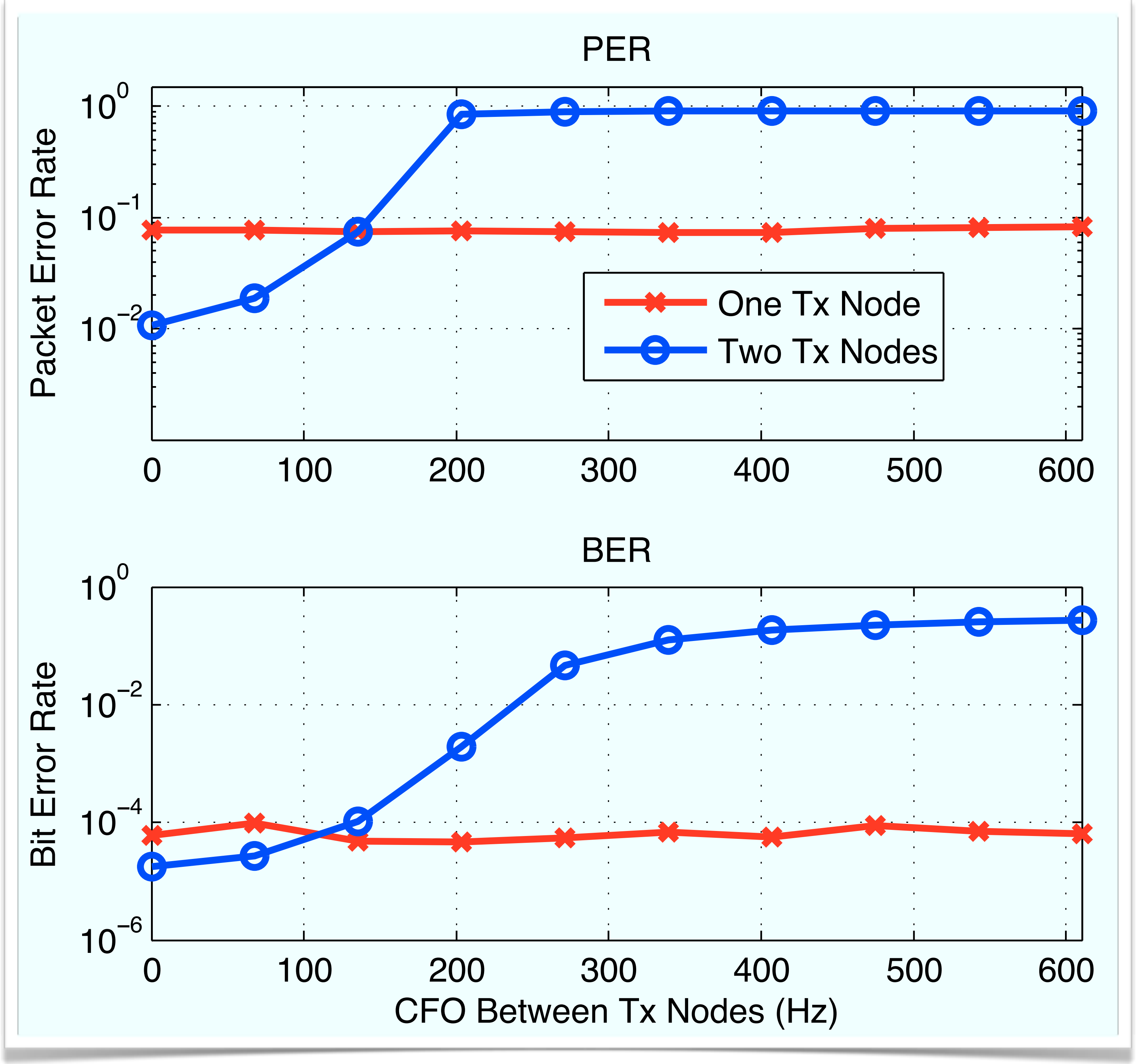
# Carrier Frequency Offset

*CFO in Decode and Forward*



- Inter-Tx CFO in DF is harder
- How bad?

# Carrier Frequency Offset

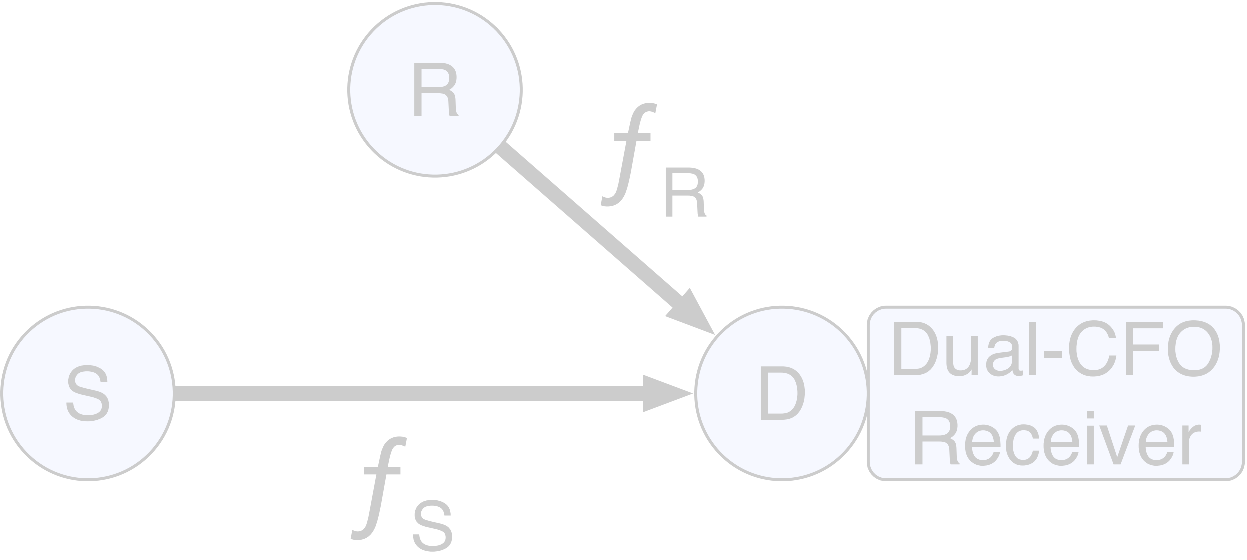


- *Experimental results*
- *1412 byte 16-QAM packets*
- *Frequency flat fading, high average SNR*

# Carrier Frequency Offset

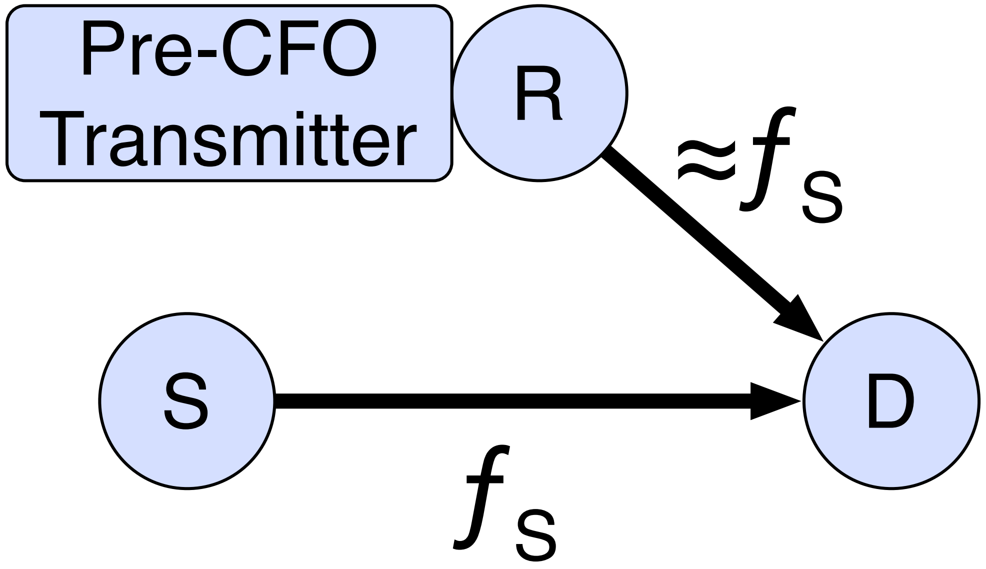
*CFO in Decode and Forward*

## Fix it at the Destination



- Simple relay
- *Really* complicated destination
  - Requires lots of estimation, overhead

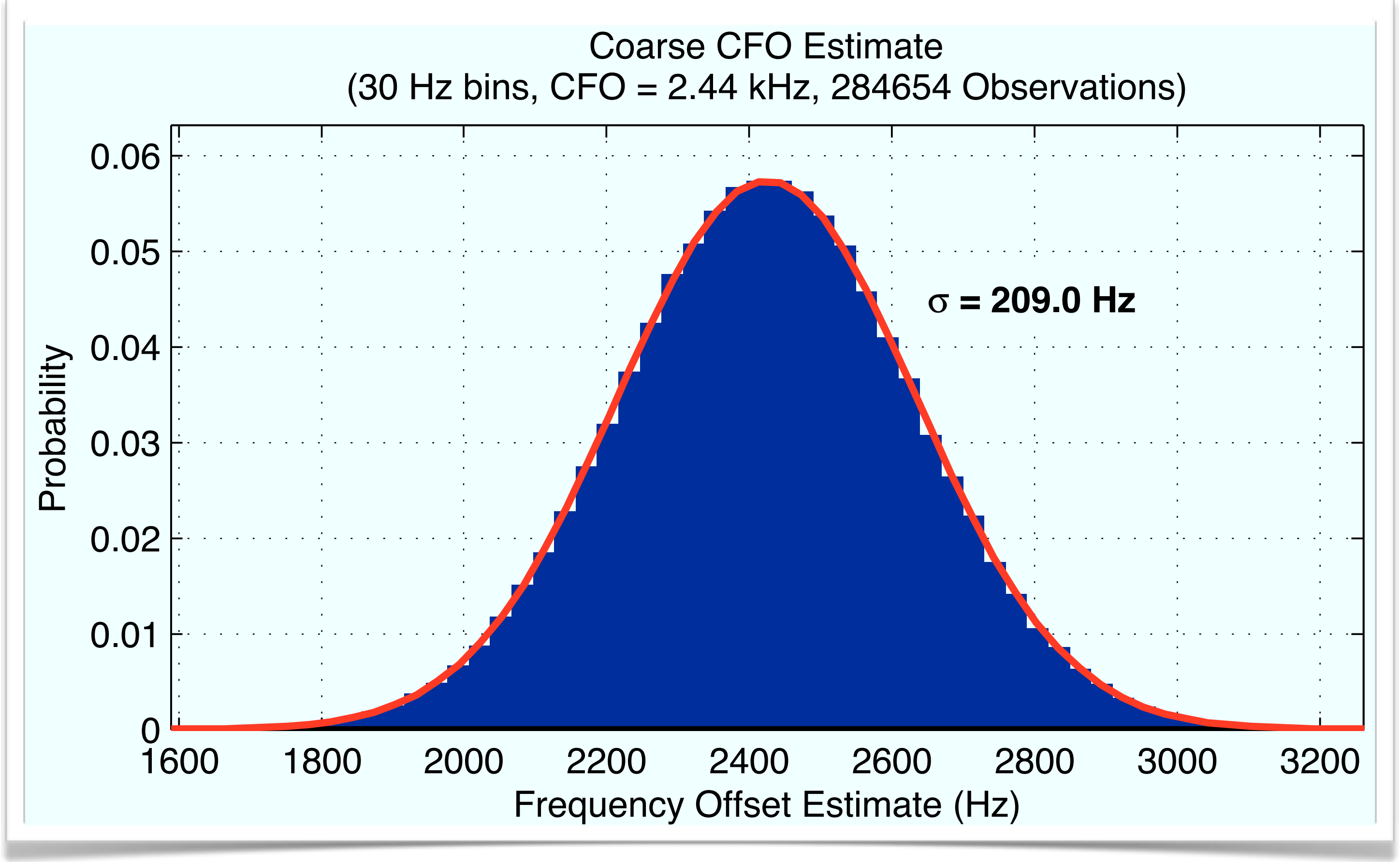
## Fix it at the Relay



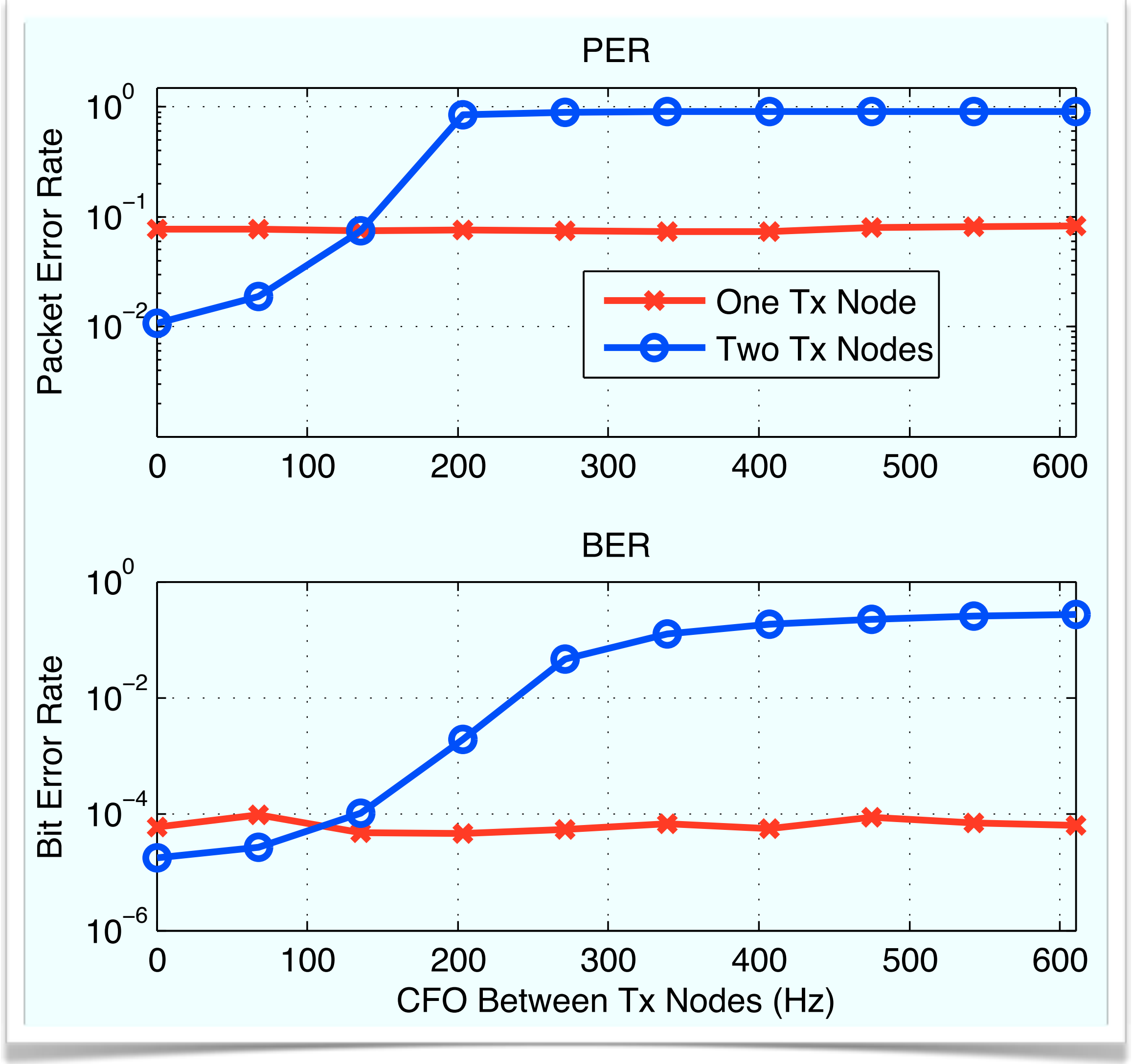
- Simple destination
- Straightforward relay
  - Pre-spin by $\Delta f_{SR}$ is easy
  - Requires good $\Delta f_{SR}$ estimate

# Carrier Frequency Offset

*CFO in Decode and Forward*



*Time Domain CFO Estimate*

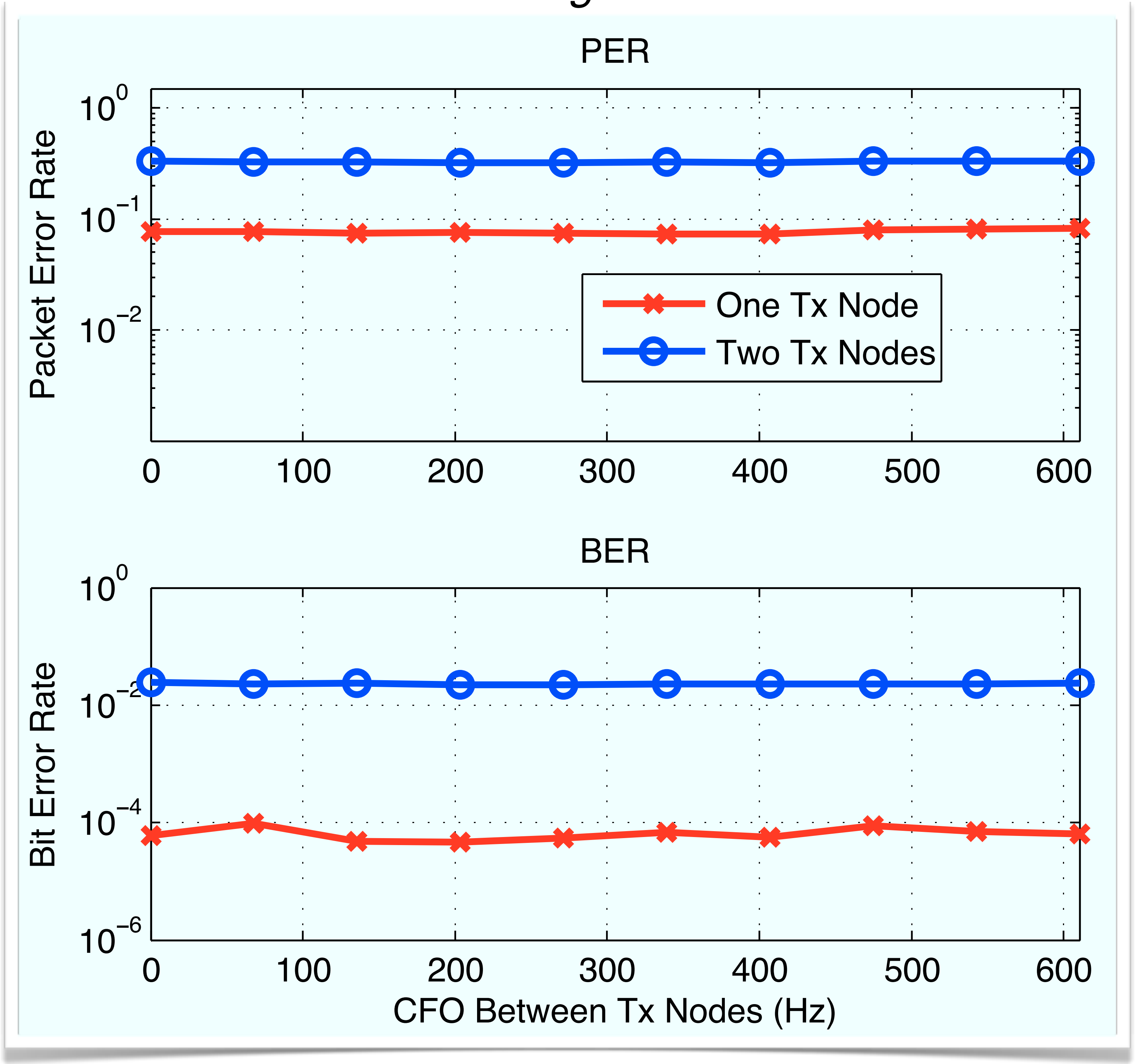


*Inter-Tx CFO Impact*

Probably not going to work...

# Carrier Frequency Offset

*CFO Pre-Correction using Time Domain Estimate*

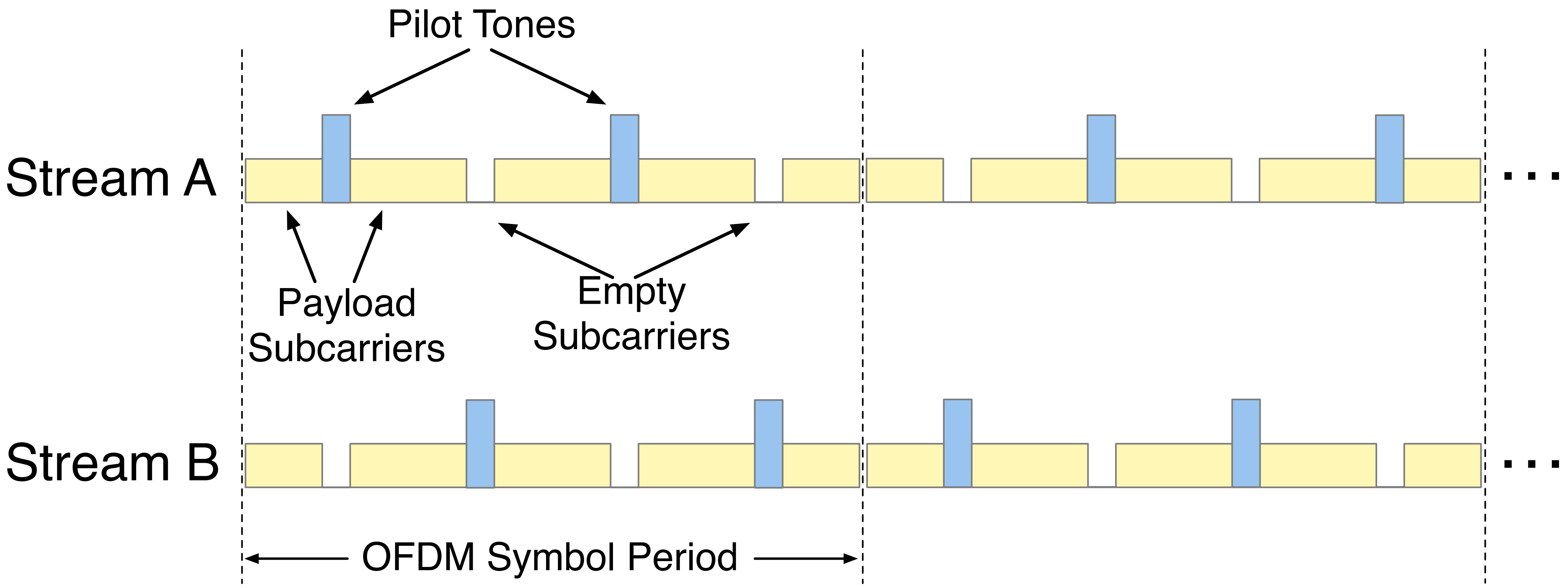


✗ Fail!

How to better estimate CFO?

# Carrier Frequency Offset

*CFO in Decode and Forward*



Pilot Tones

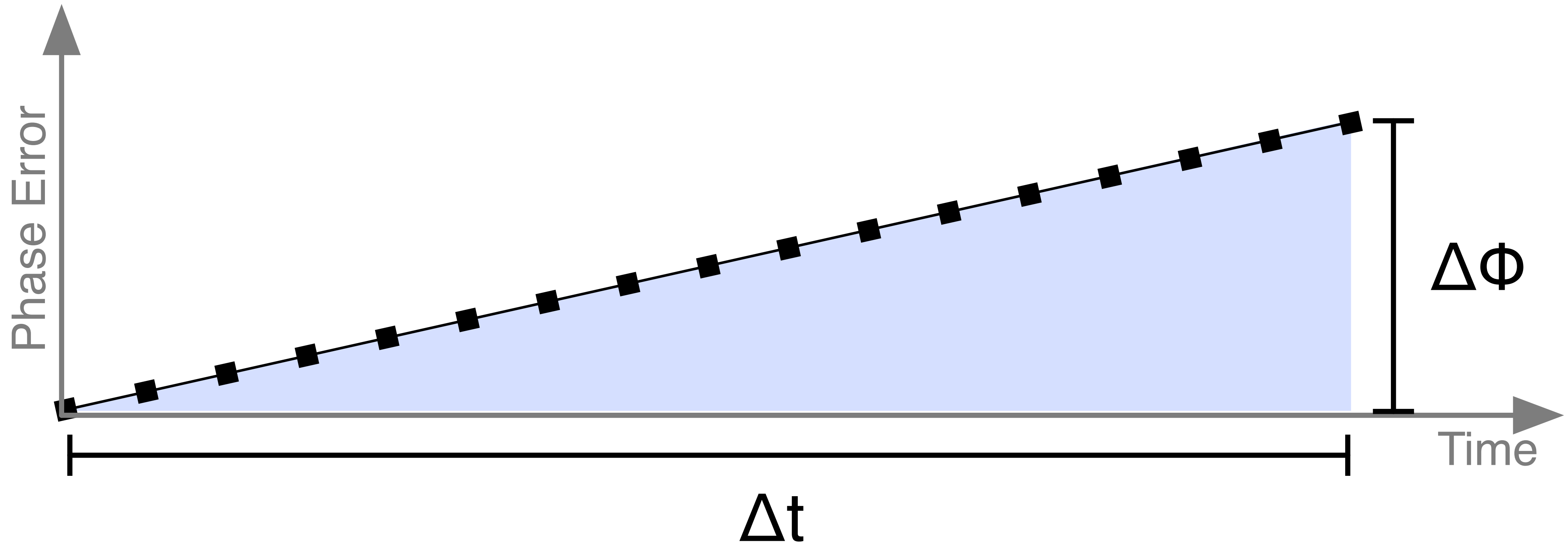
Stream A

Payload Subcarriers

Empty Subcarriers

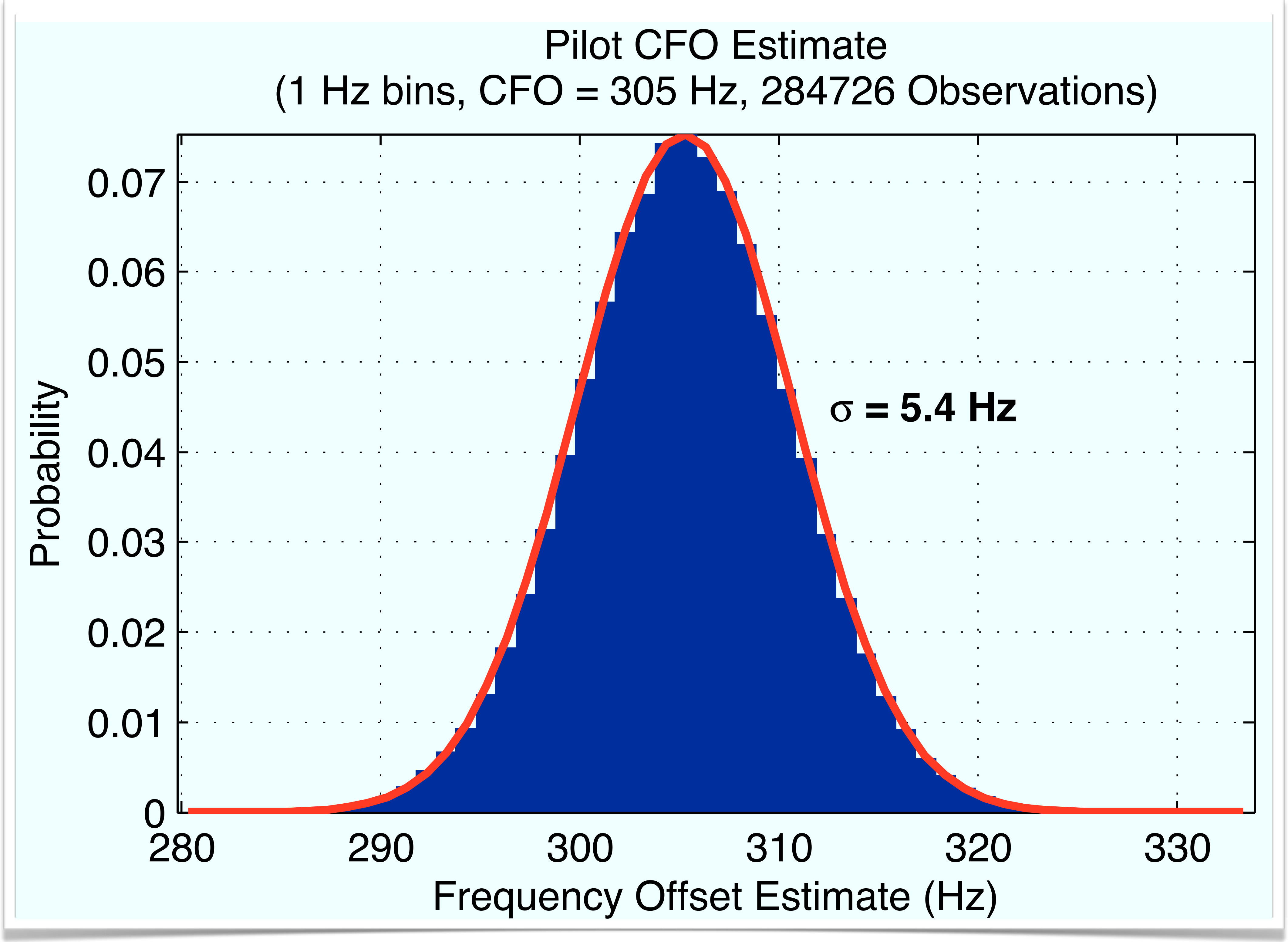
Stream B

OFDM Symbol Period



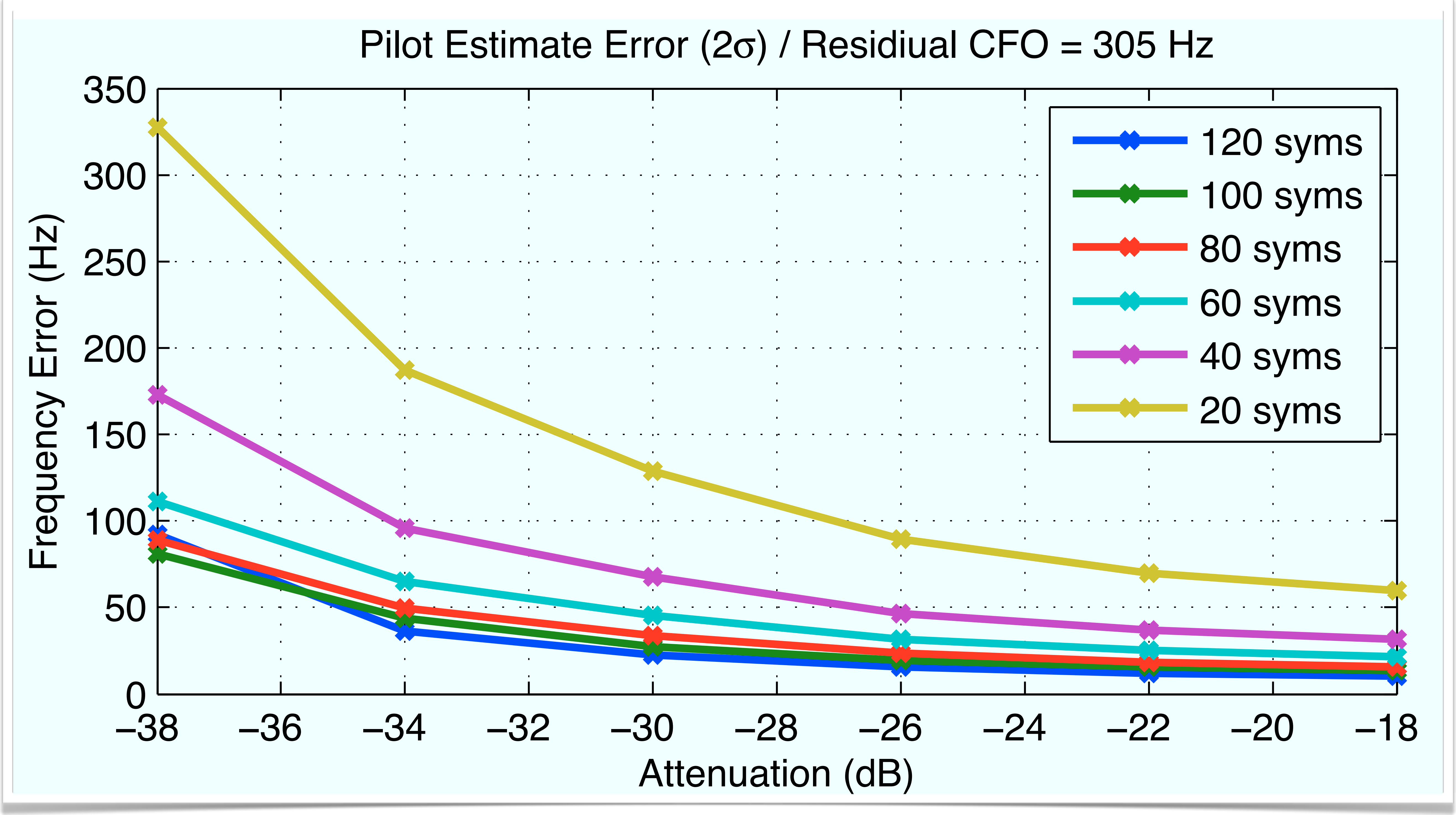
# Carrier Frequency Offset

*Frequency Domain CFO Estimator*



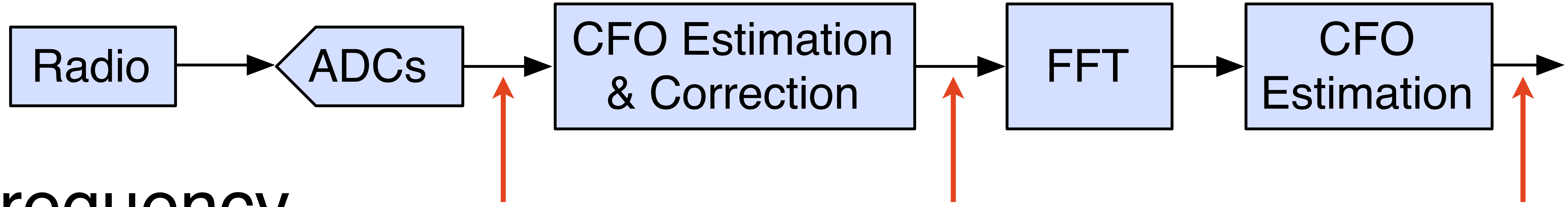
# Carrier Frequency Offset

*Frequency Domain CFO Estimator*



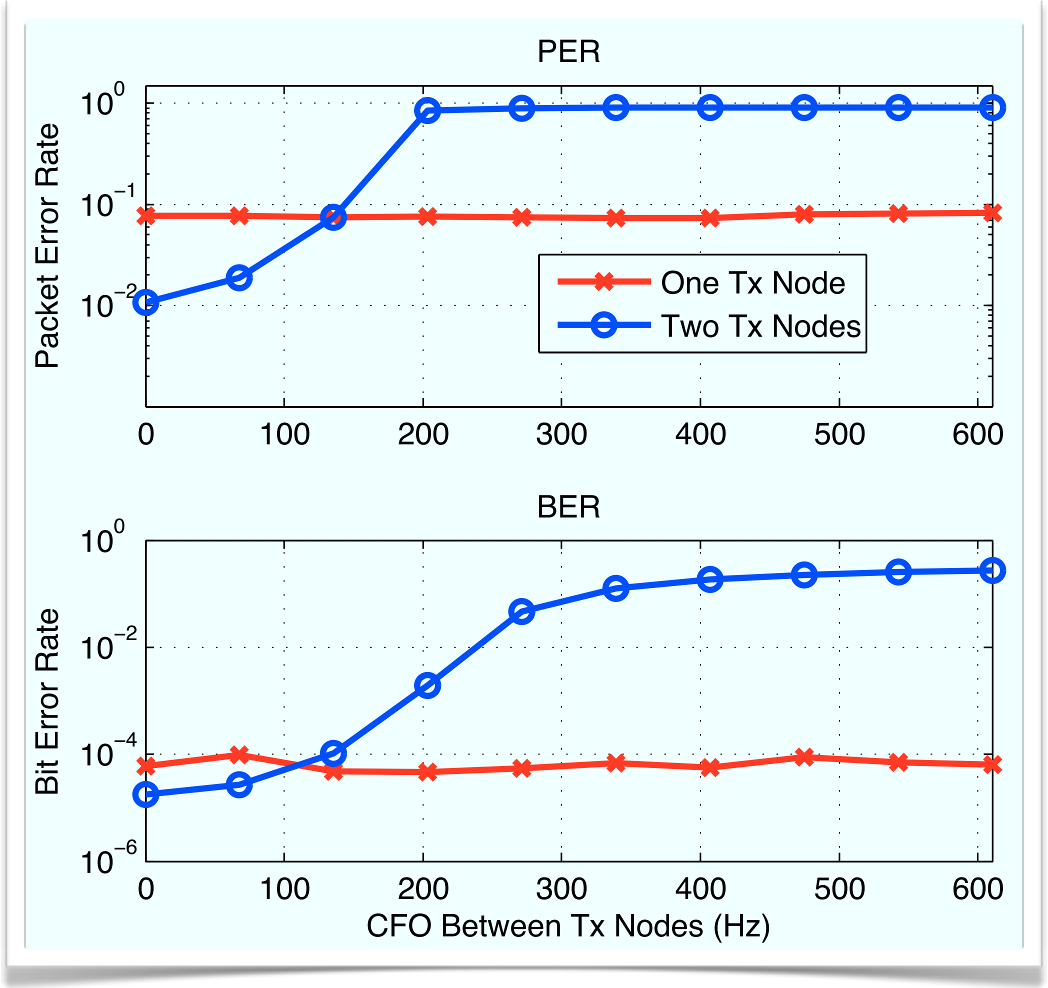
- Experimental results
- AWGN, SISO OFDM

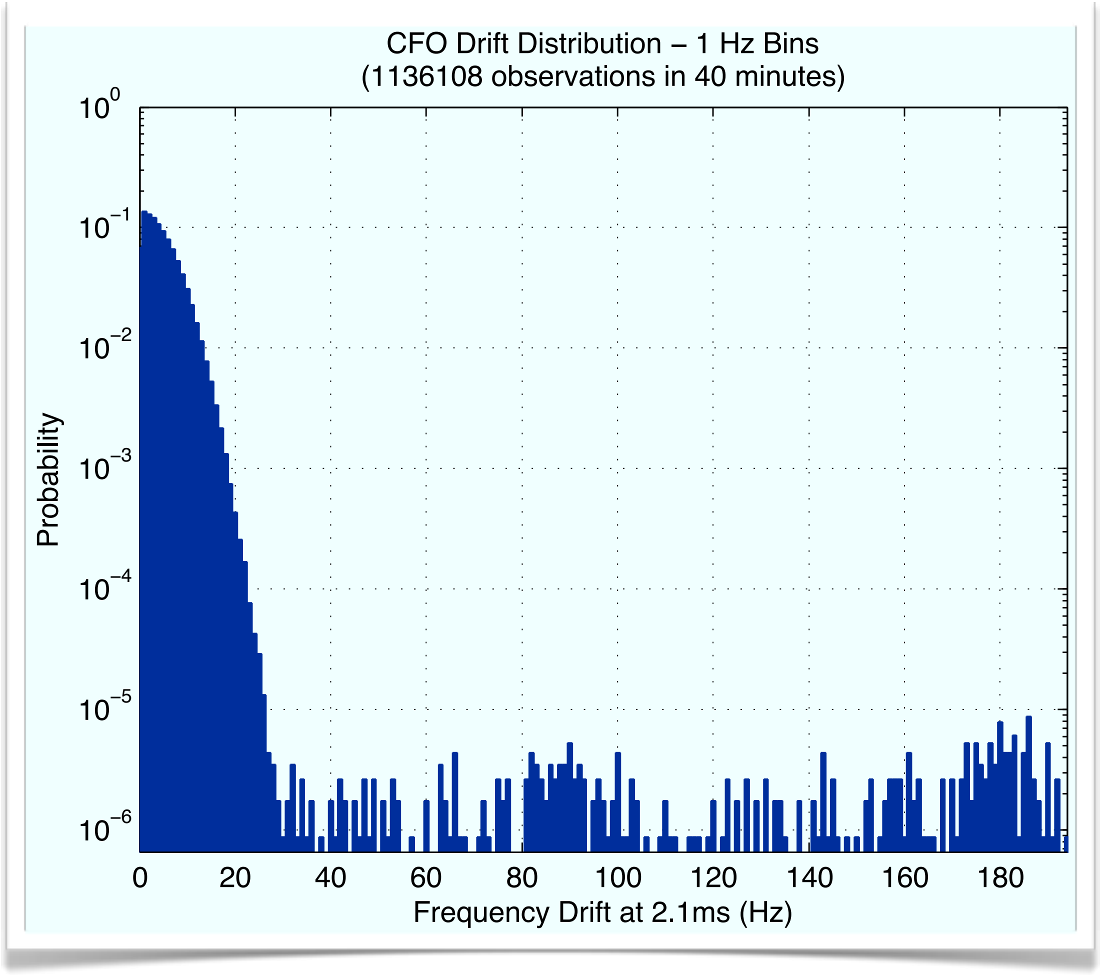
# Carrier Frequency Offset



Radio → ADCs → CFO Estimation & Correction → FFT → CFO Estimation →

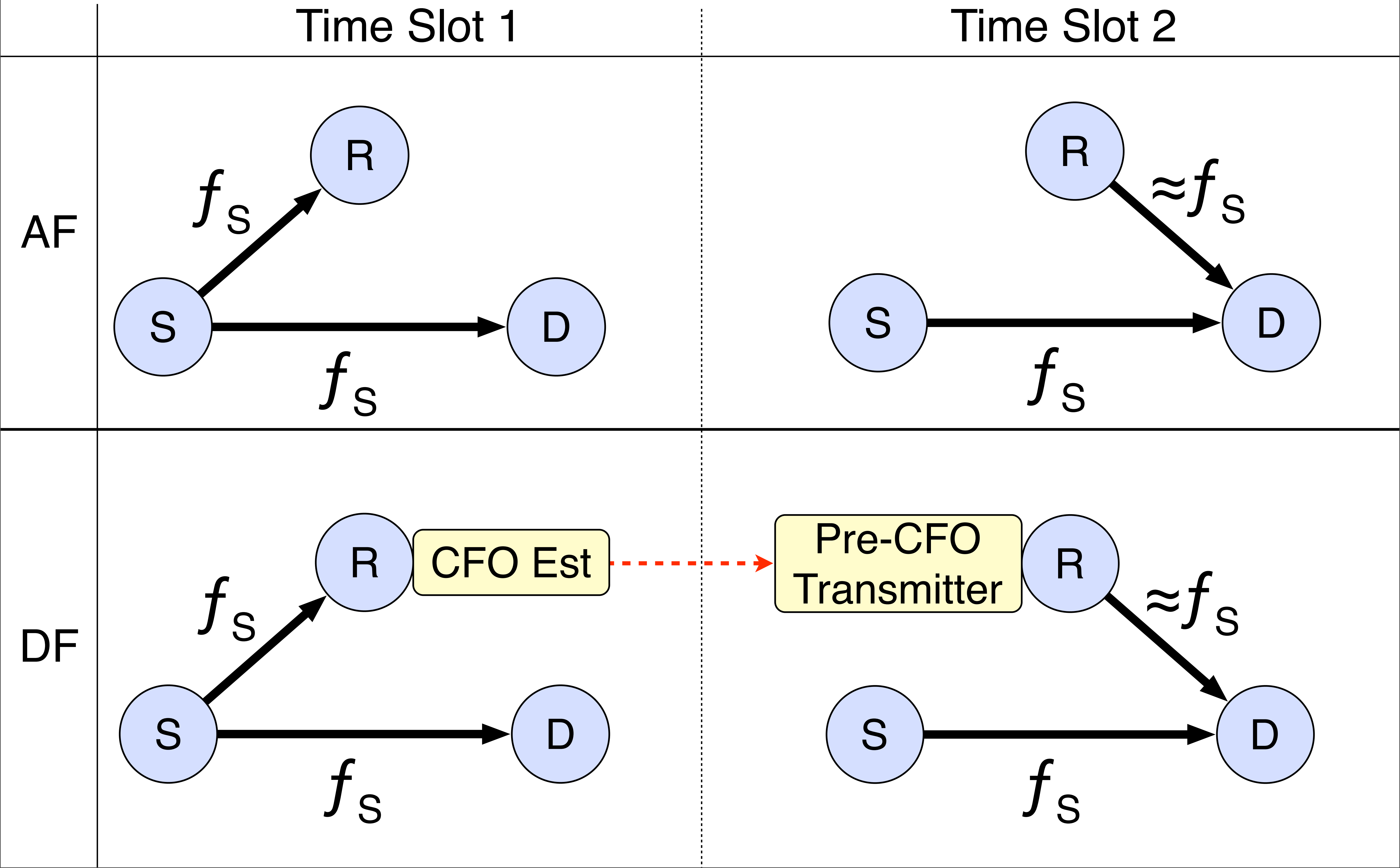
Frequency Error: < 20 kHz ≈500 Hz ≈11 Hz





✓ Success!

# CFO and Cooperation



# Outline

- Brief background
- Carrier frequency offsets
- **Experiment design**
- Characterization results
- Future work

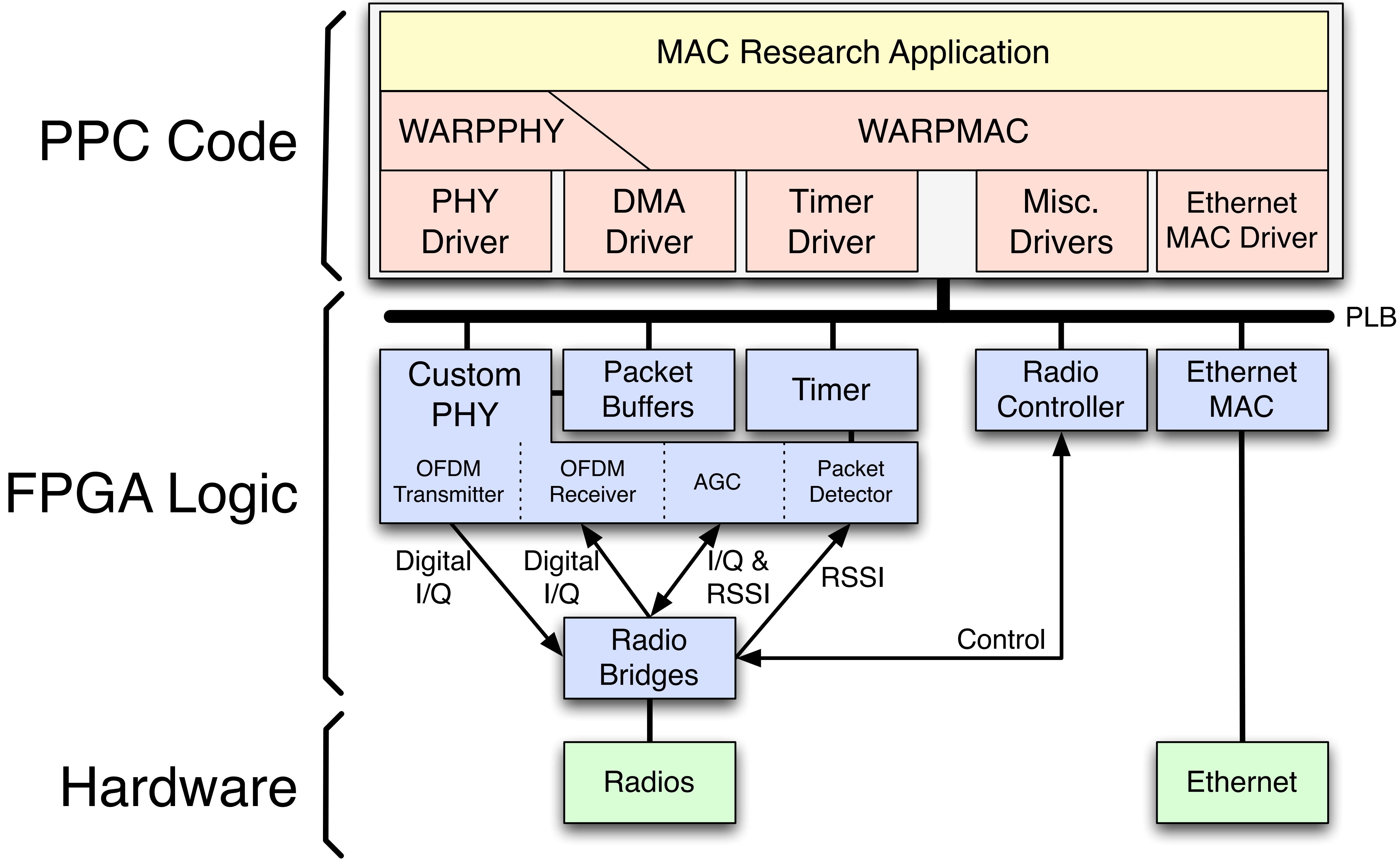
# Experiment Design

- How to integrate everything?
- How to control propagation?
- What variables to sweep?
- What to measure?
- How to coordinate everything?

# Experiment Design

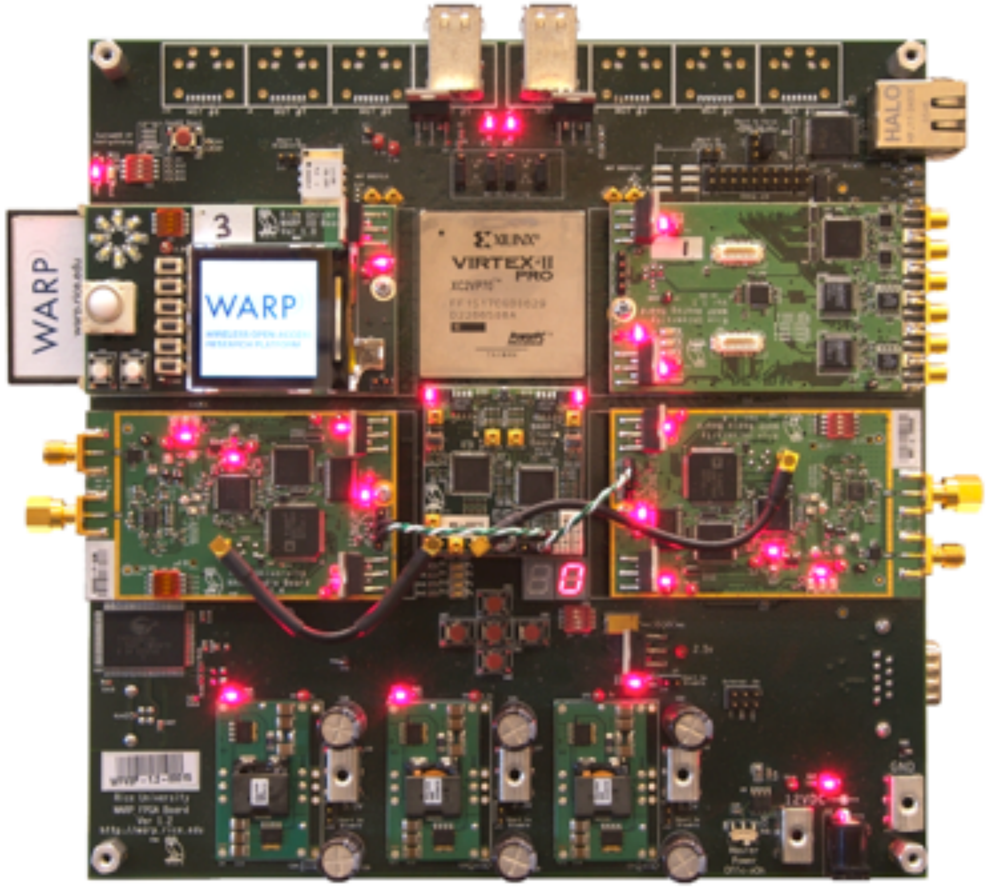
- How to integrate everything?
- How to control propagation?
- What variables to sweep?
- What to measure?
- How to coordinate everything?

# Design Integration



# Design Integration

- One design implements everything
    - OFDM Tx/Rx processing
    - Source/Relay/Destination roles
    - NC/AF/DF per-packet
    - MAC protocol support



- Already posted on WARP site as OFDM Reference Design v15

# Design Integration

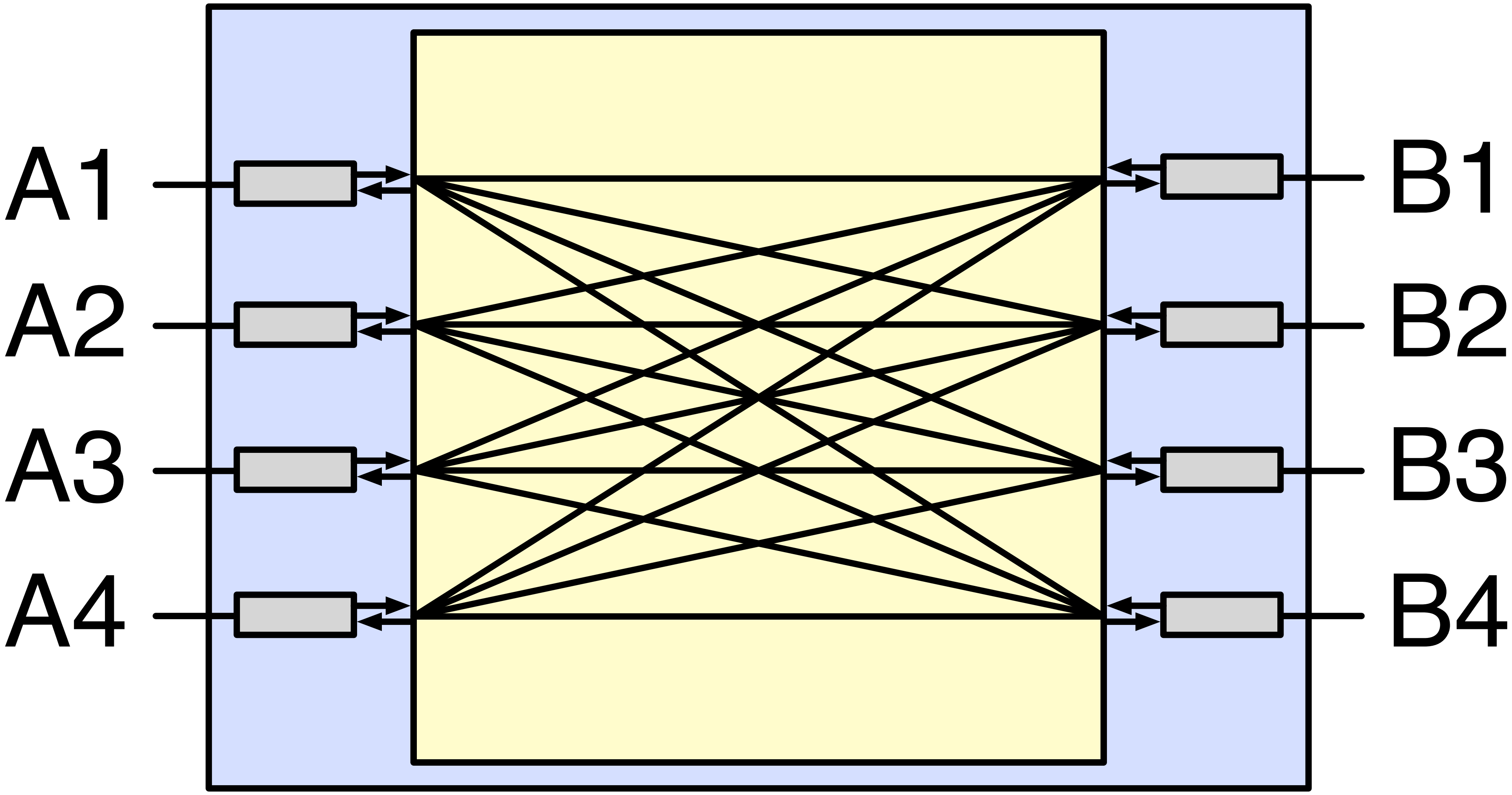
| FPGA Resource | PHY Tx | PHY Rx | Full Design | Available (XC2VP70) |
|---|---|---|---|---|
| Logic Slices | 4512 | 10360 | 29182 | 33088 |
| Multipliers | 88 | 122 | 214 | 328 |
| Block RAMs | 8 | 140 | 309 | 328 |
| I/O | – | – | 548 | 964 |

# Experiment Design

- How to integrate everything?
- How to control propagation?
- What variables to sweep?
- What to measure?
- How to coordinate everything?

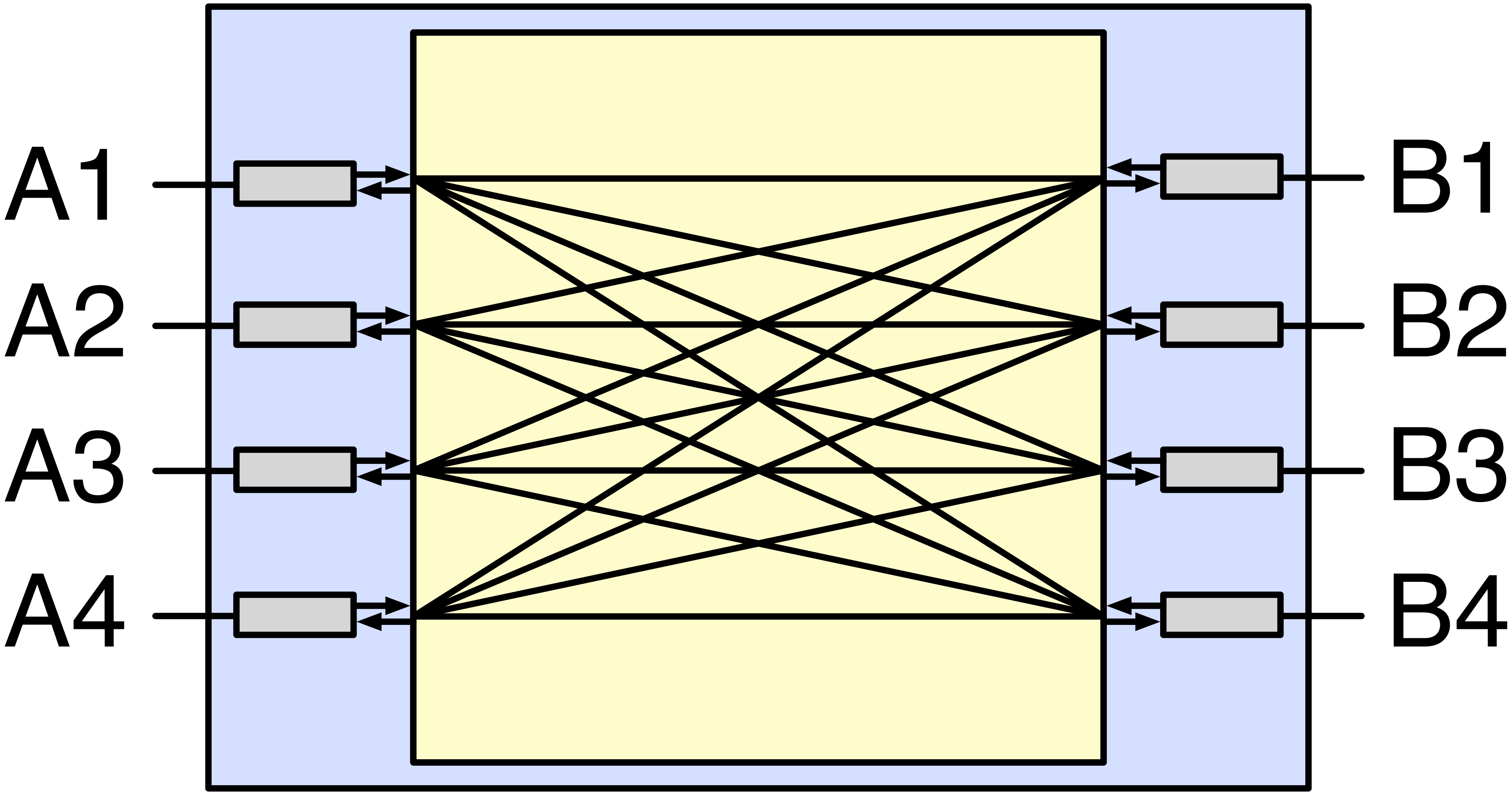
# Channel Emulator

*Azimuth ACE 400WB*



# Channel Emulator

*Azimuth ACE 400WB*

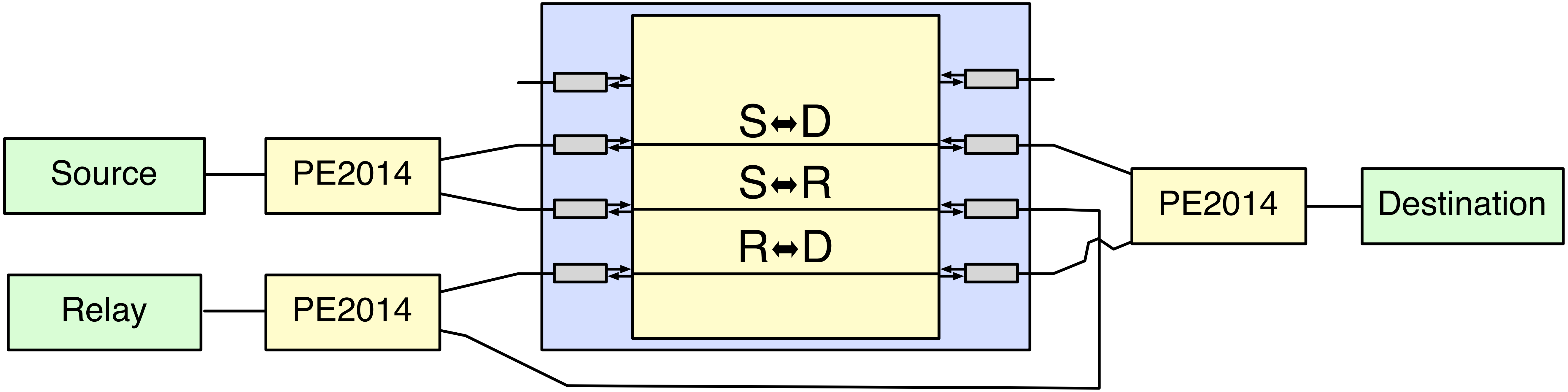


Two limitations:

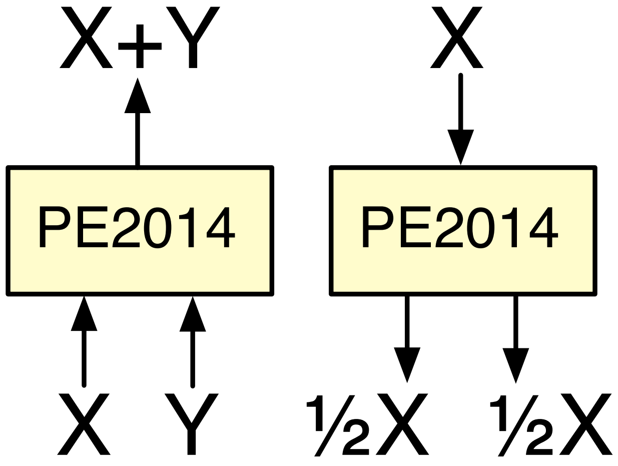
- One channel model per output
- Only constructs A↔B paths

# Channel Emulator

*Cooperative Experiment Connections*



- Three independent paths
- All nodes inter-connected



# Channel Emulator

*Cooperative Experiment Connections*

**WARP Nodes**

Source

Relay

Destination

RF Cabling

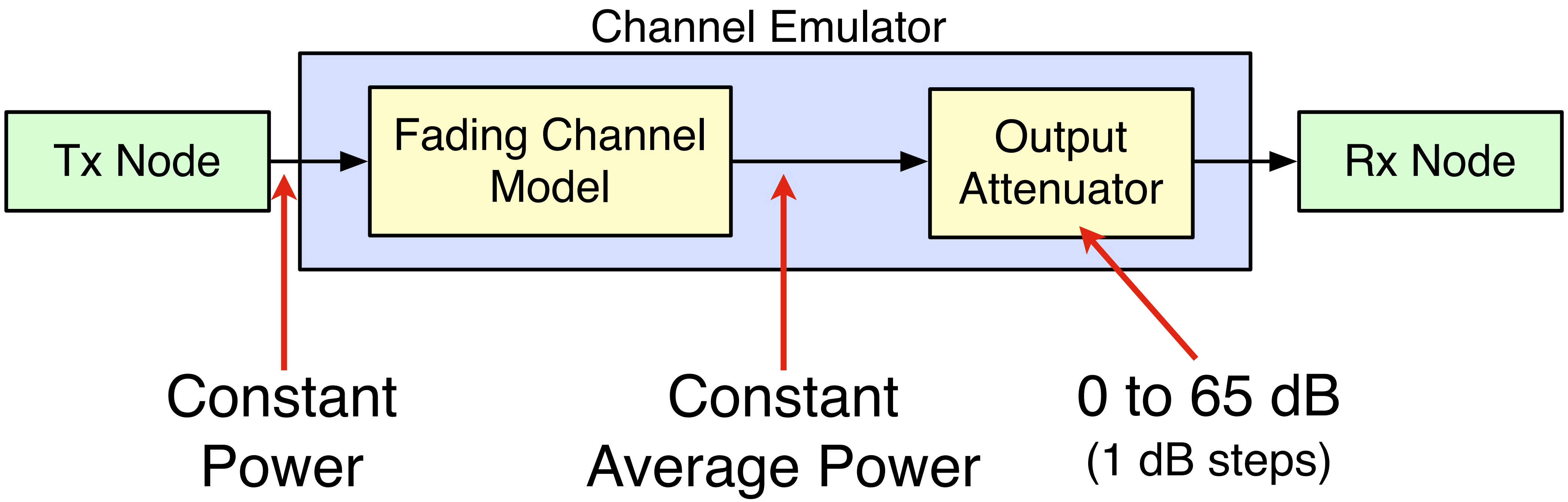
Channel Emulator

*Cheesy photo © 2010 by Chris Hunter*

# Experiment Design

- How to integrate everything?
- How to control propagation?
- What variables to sweep?
- What to measure?
- How to coordinate everything?

# Topologies



Channel Emulator

Tx Node

Fading Channel Model

Output Attenuator

Rx Node

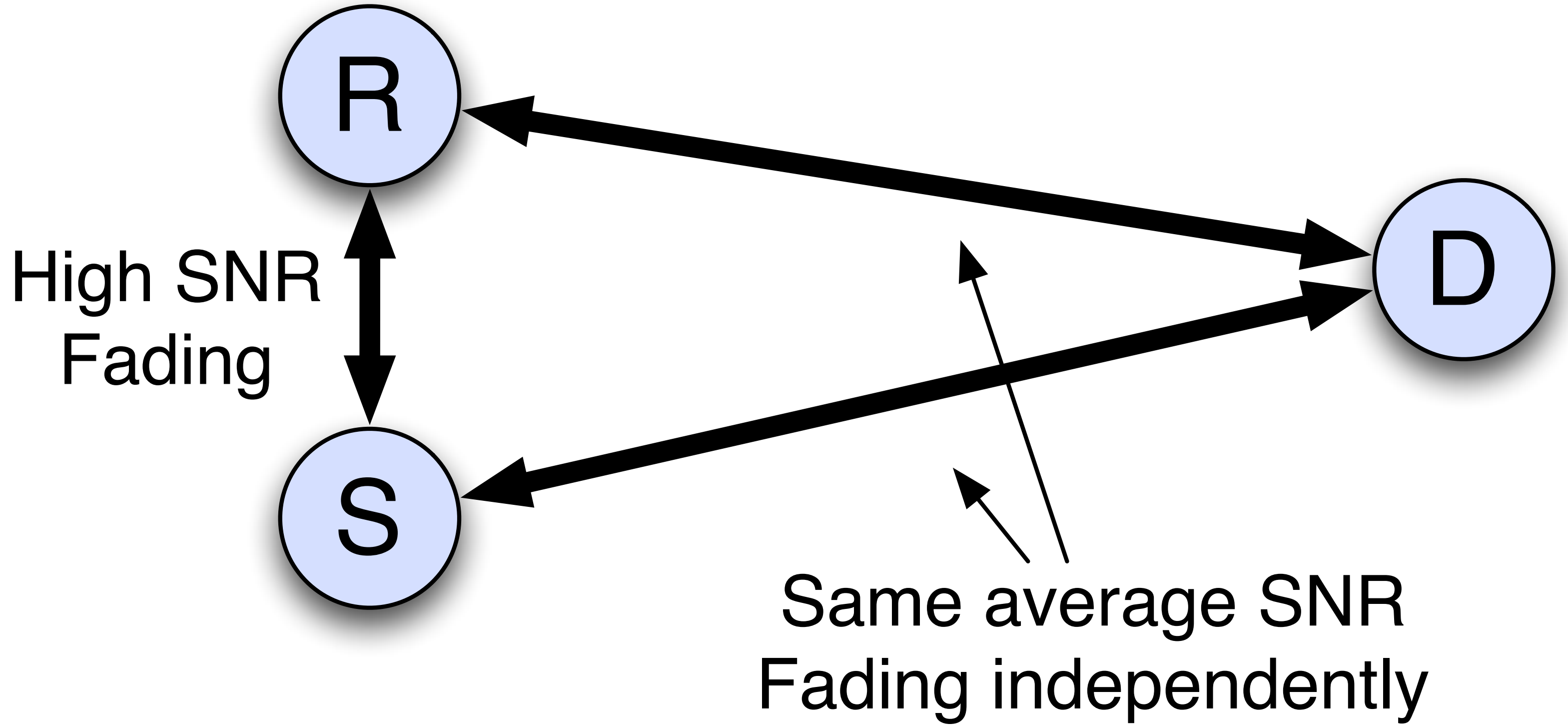
Constant Power

Constant Average Power

0 to 65 dB
(1 dB steps)

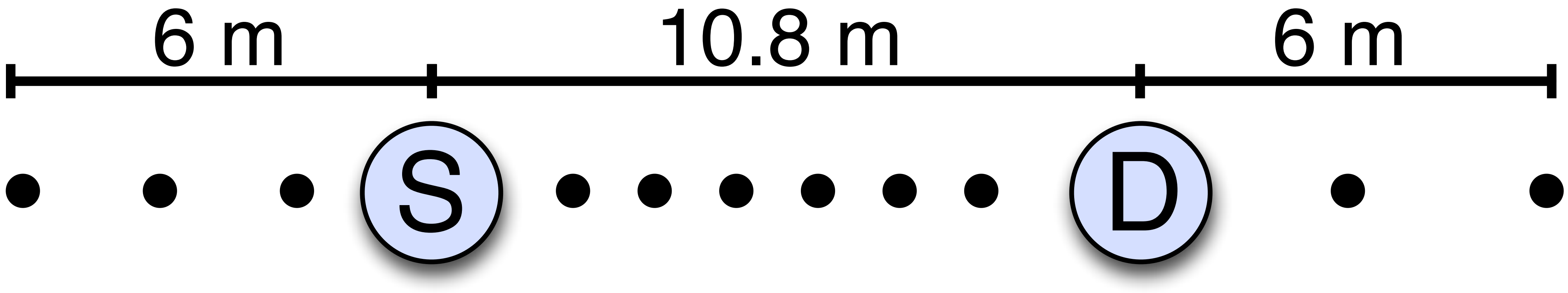
- Performance depends on relative node distances
- Emulate distances with average path loss
- Controlled only via emulator - **no change to nodes**
- Full sweep of $66^3$ attenuation combinations infeasible

# Co-located Source/Relay



- All links are fading
- SR is fixed at high average SNR
- Parameterized by common SD/RD average SNR
- Interesting usage case for cooperation

# Linear Topology



● = Relay Locations

- All links are fading
- SD link is fixed at moderate average SNR
- Parameterized by relay location along SD line

# Experiment Design

- How to integrate everything?
- How to control propagation?
- What variables to sweep?
- What to measure?
- How to coordinate everything?

# Metrics

## Packet Error Rate

$N_{Tx}$: Number of packets transmitted

$N_{RxGood}$: Number of packets received with no bit errors

$$PER = \frac{N_{Tx} - N_{RxGood}}{N_{Tx}}$$

---

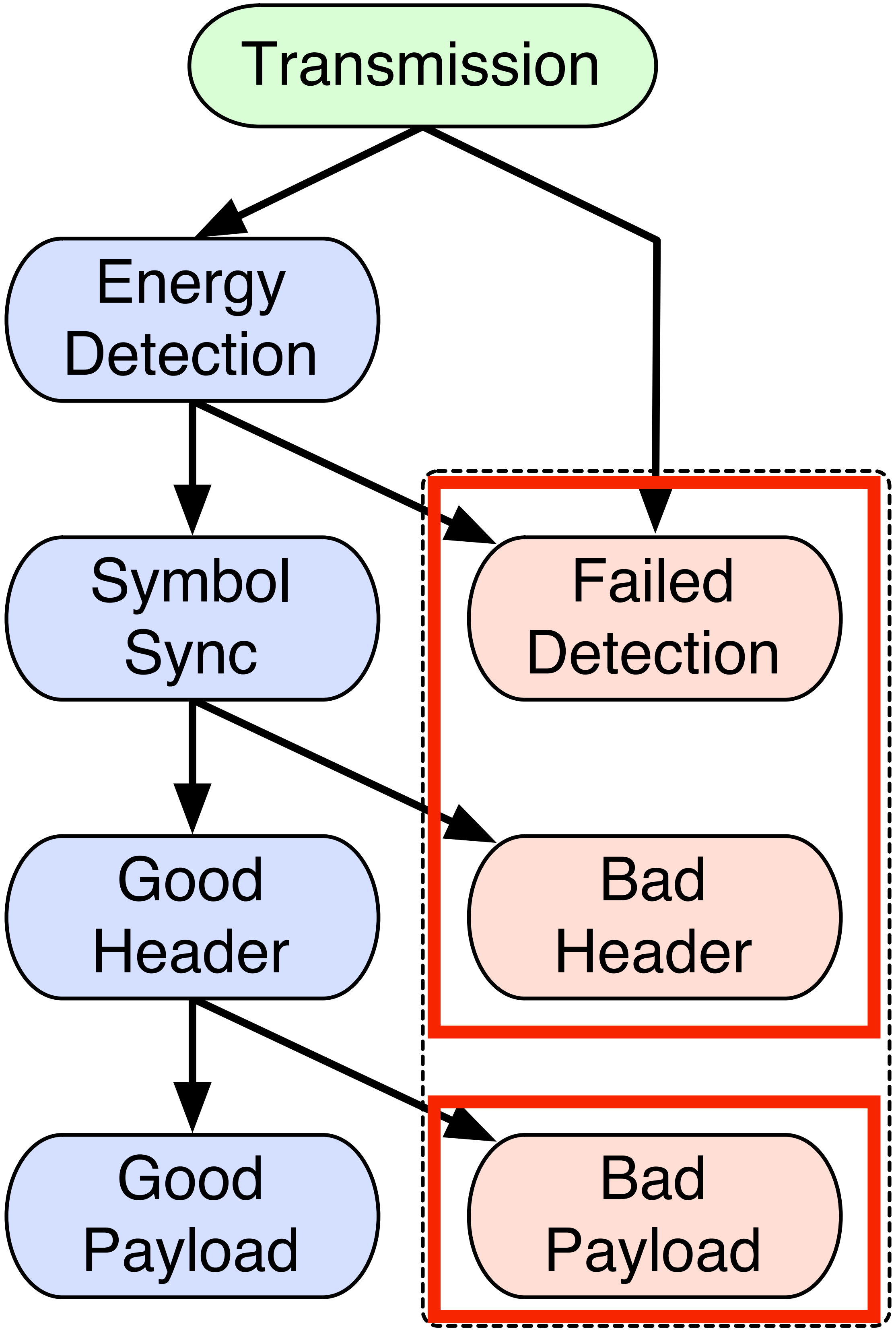
## Bit Error Rate

$B_{Error}$: Number of bit errors observed

$B_{Total}$: Number of bits processed

$$BER = \frac{B_{Error}}{B_{Total}}$$

# Reception Outcomes

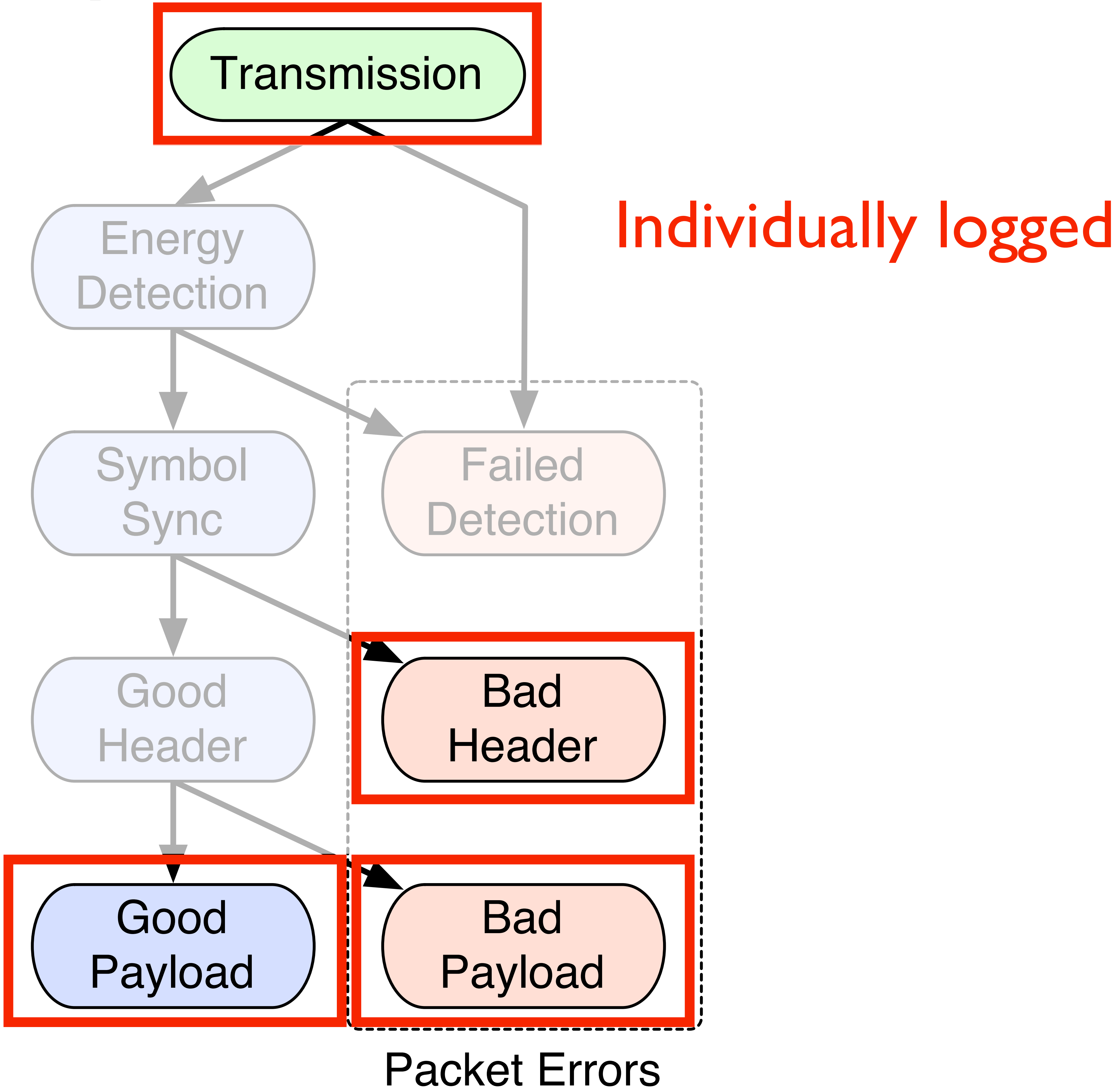


Packet Errors

Often not considered

Only state with bit errors

# Reception Outcomes

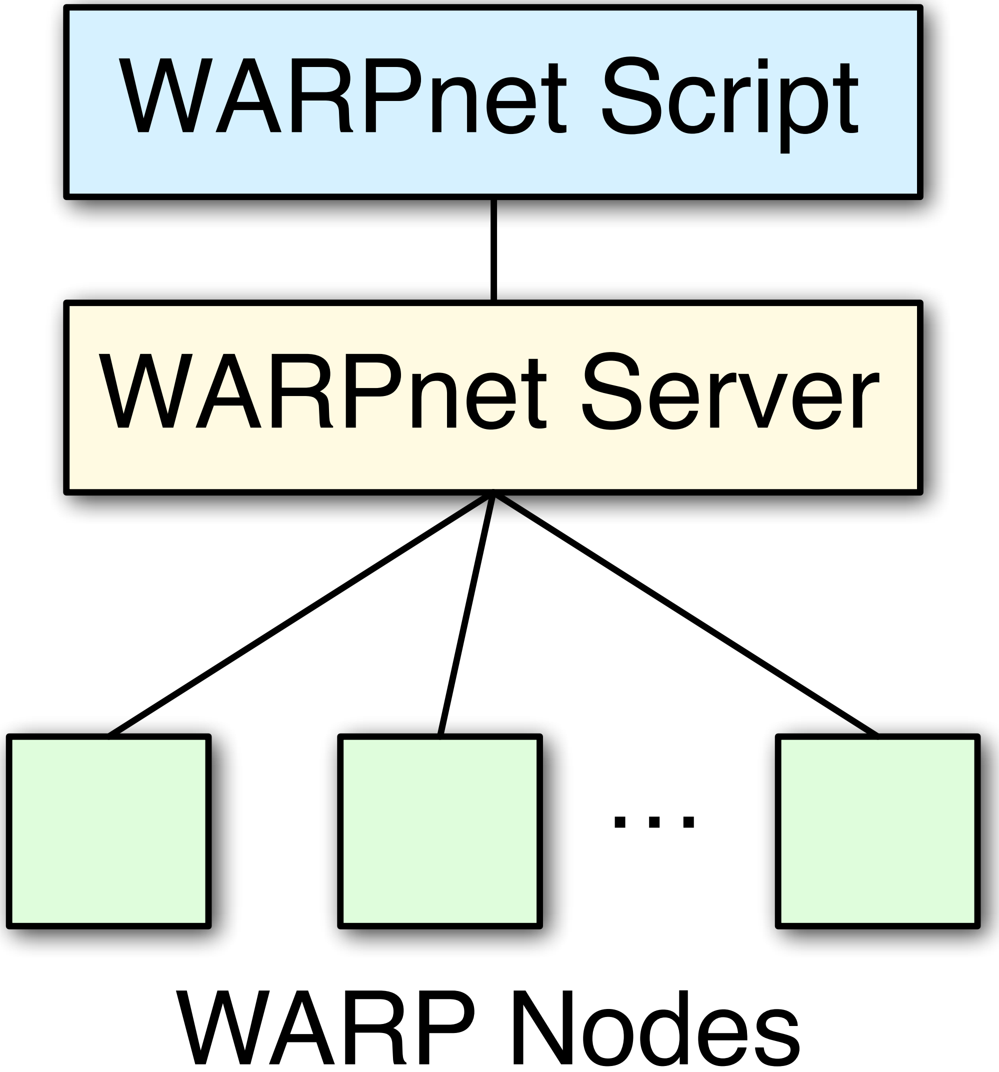


# Experiment Design

- How to integrate everything?
- How to control propagation?
- What variables to sweep?
- What to measure?
- How to coordinate everything?

# WARPnet Framework

- Automated testing for WARP nodes
- Abstraction between script and nodes
- All custom Python and open-source
- Thanks Sid!



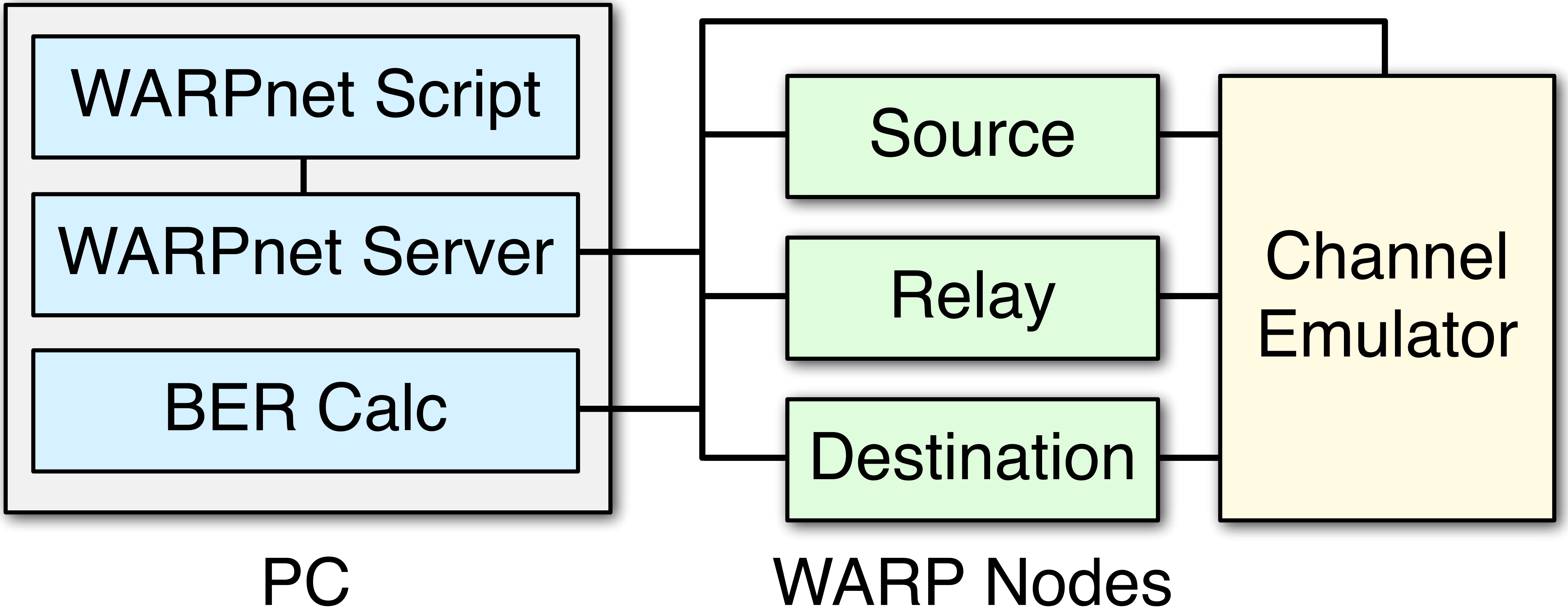
WARPnet Script

WARPnet Server

...

WARP Nodes

# WARPnet Framework



- BER co-processor mimics a node
- Gathers PER/BER in one experiment
- One Python script controls everything

# Experiment Design

| | |
|---|---|
| How to integrate everything? | ✓ Single FPGA design |
| How to control propagation? | ✓ Channel emulator |
| What variables to sweep? | ✓ Attenuations for interesting topologies |
| What to measure? | ✓ Bit & packet errors |
| How to coordinate everything? | ✓ WARPnet |

# Experimental Results

- Iterated on every combination of:
  - [isoTri, eqTri, lin10, lin18] topologies
  - [QPSK, 16-QAM] modulation
  - [1412, 692] byte payloads
  - Frequency [flat, selective] fading
  - [NC, AF, DF] schemes

    (plus [MHOP, AFGH] in linear topologies)

# Experimental Results

- Iterated on every combination of:
  - 10 × 4 [isoTri, eqTri, lin10, lin18] topologies
  - × 2 [QPSK, 16-QAM] modulation
  - × 2 [1412, 692] byte payloads
  - × 2 Frequency [flat, selective] fading
  - × 3 [NC, AF, DF] schemes

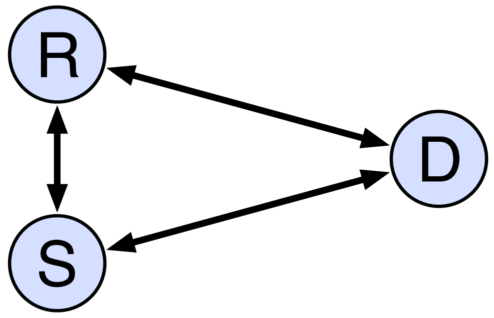
    (plus [MHOP, AFGH] in linear topologies)

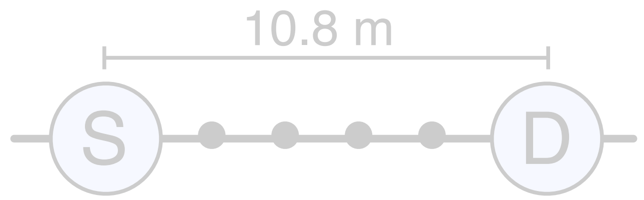
- 300+ million transmissions total

# Experimental Results

- Iterated on every combination of:
  - [isoTri, eqTri, lin10, lin18] topologies
  - [QPSK, 16-QAM] modulation
  - [1412, 692] byte payloads
  - Frequency [flat, selective] fading
  - [NC, AF, DF] schemes

    (plus [MHOP, AFGH] in linear topologies)

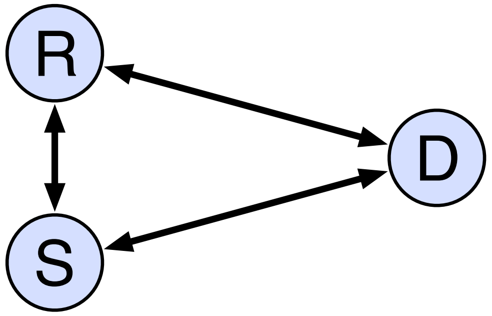


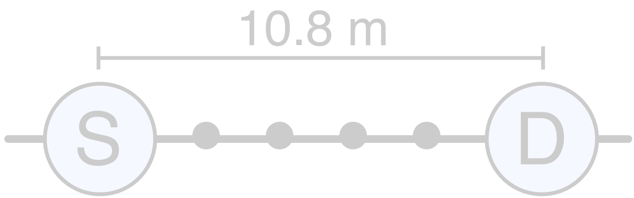


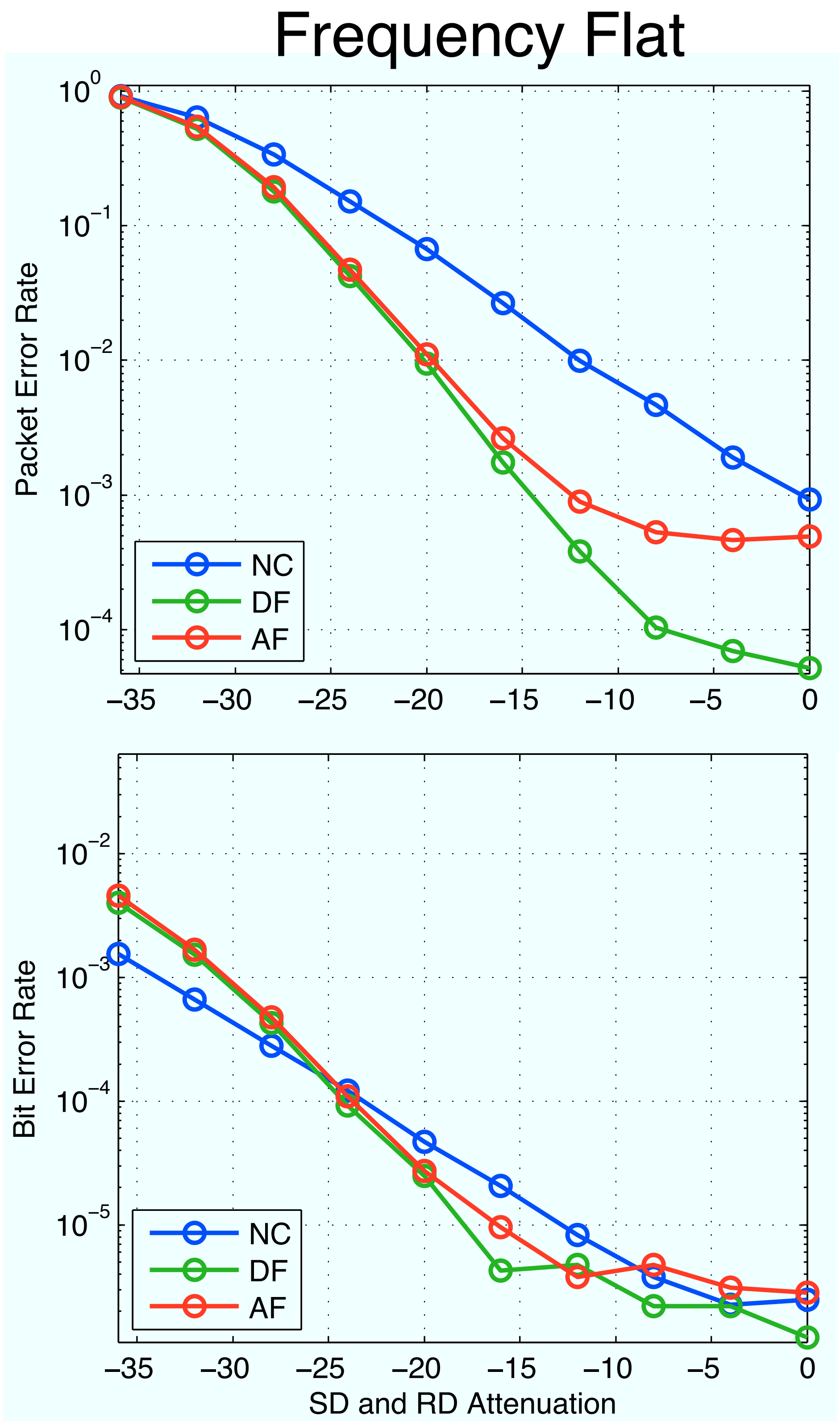
PER

BER

*QPSK*
*1412 bytes*
*6.9M packets*



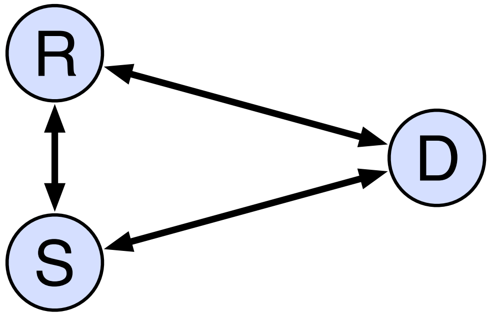


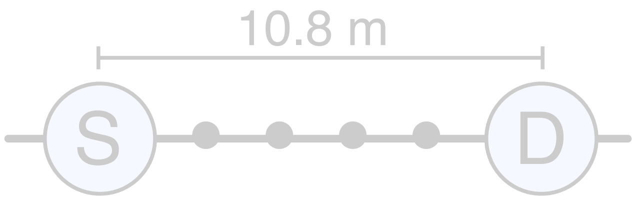


PER

BER

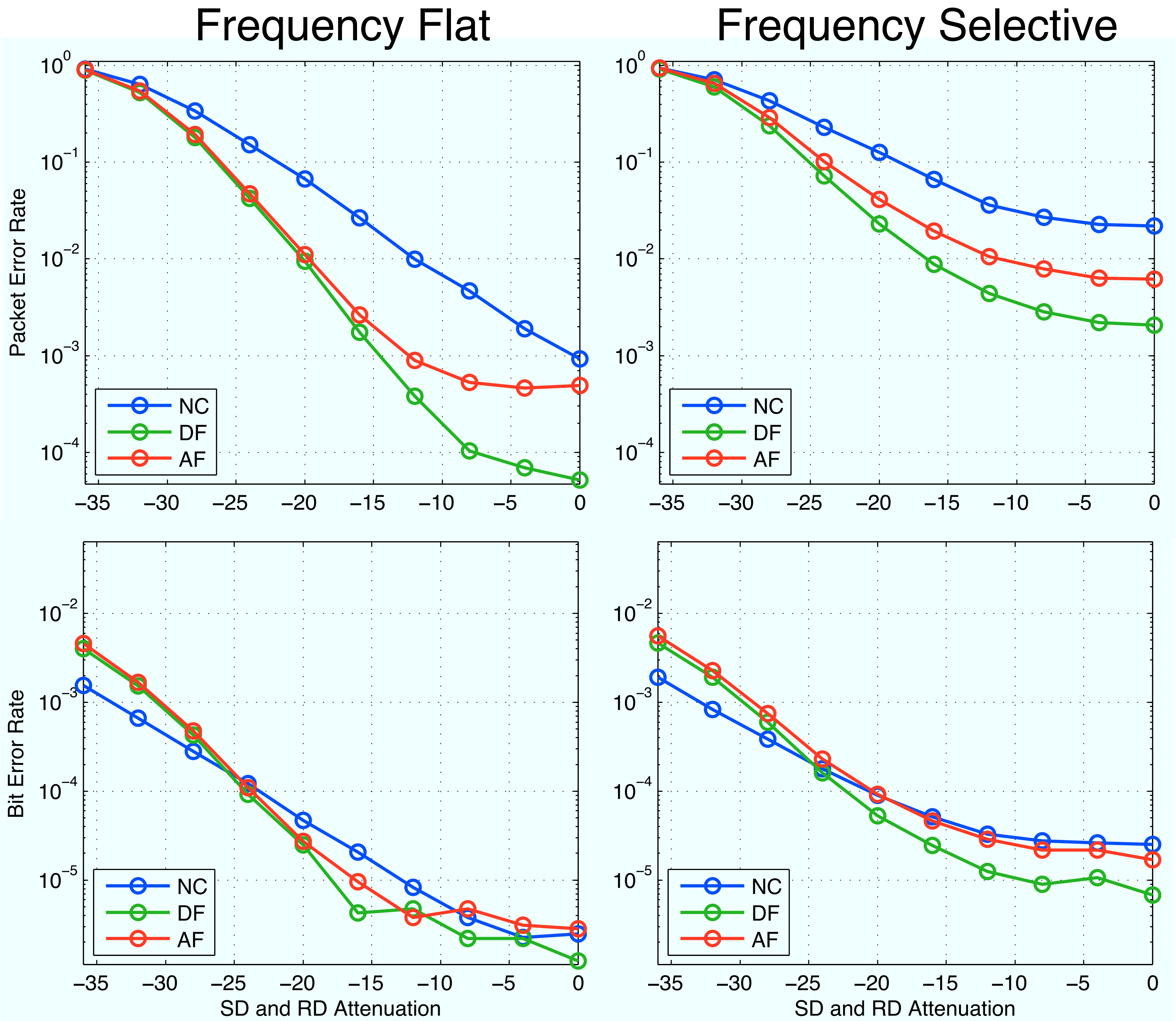
*QPSK*
*1412 bytes*
*6.9M packets*





PER

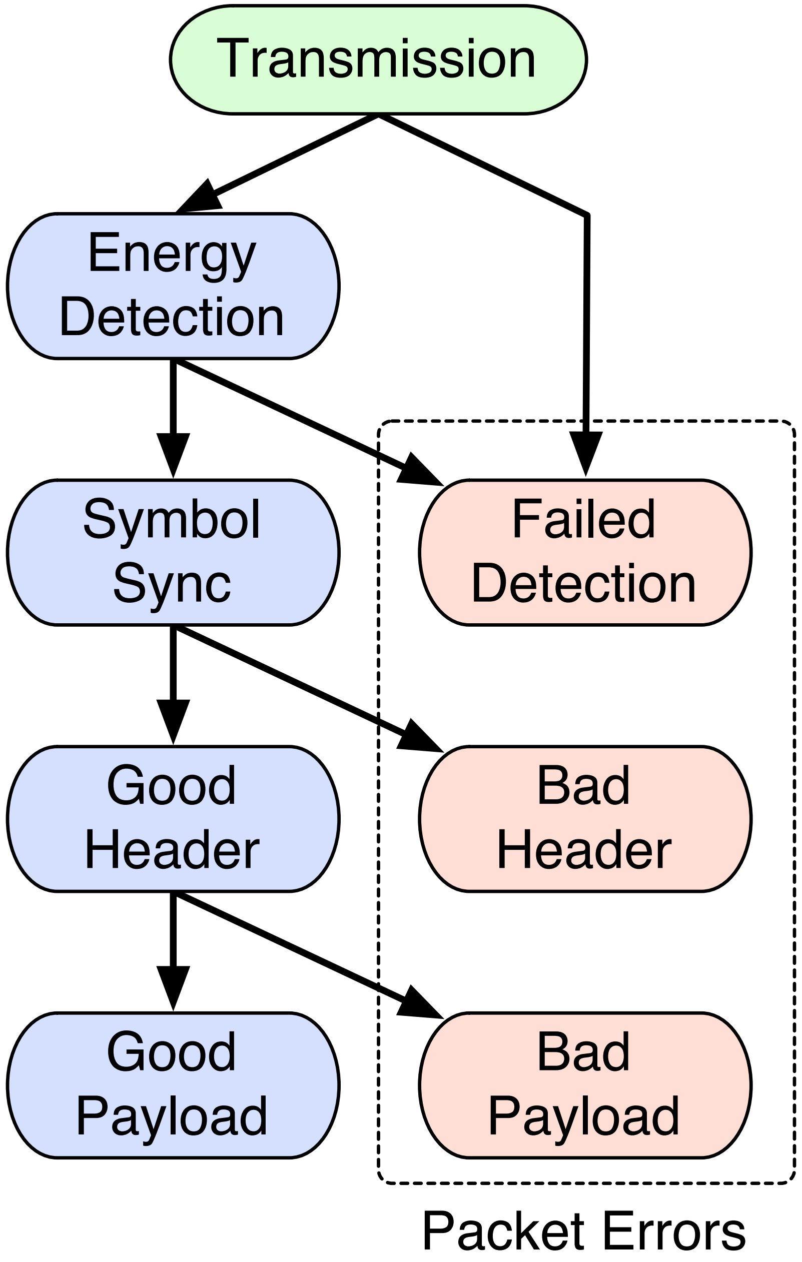
BER



*QPSK*

*1412 bytes*

*6.9M packets*



Transmission

Energy Detection

Symbol Sync

Good Header

Good Payload

Failed Detection

Bad Header

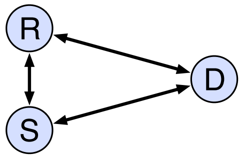
Bad Payload

Packet Errors

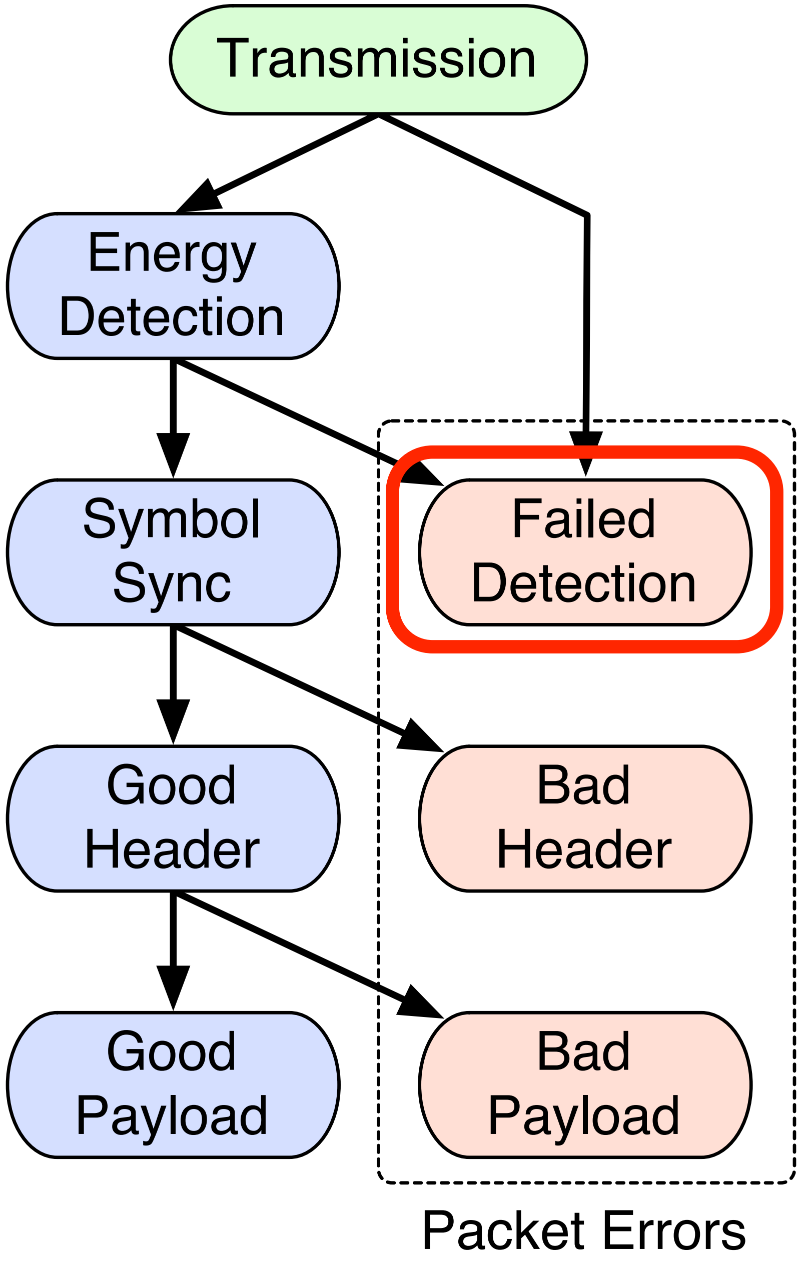
*QPSK /1412 bytes*

*Flat fading*

*6.9M packets*



## P(No Detection)



Transmission

Energy Detection

Symbol Sync

Good Header

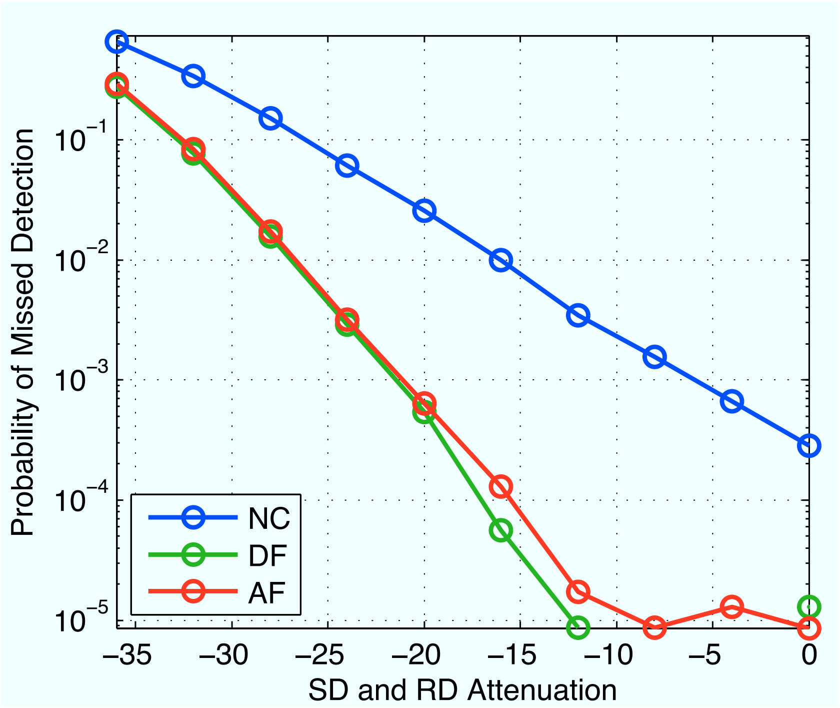
Good Payload

Failed Detection

Bad Header

Bad Payload

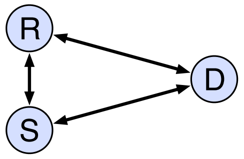
Packet Errors

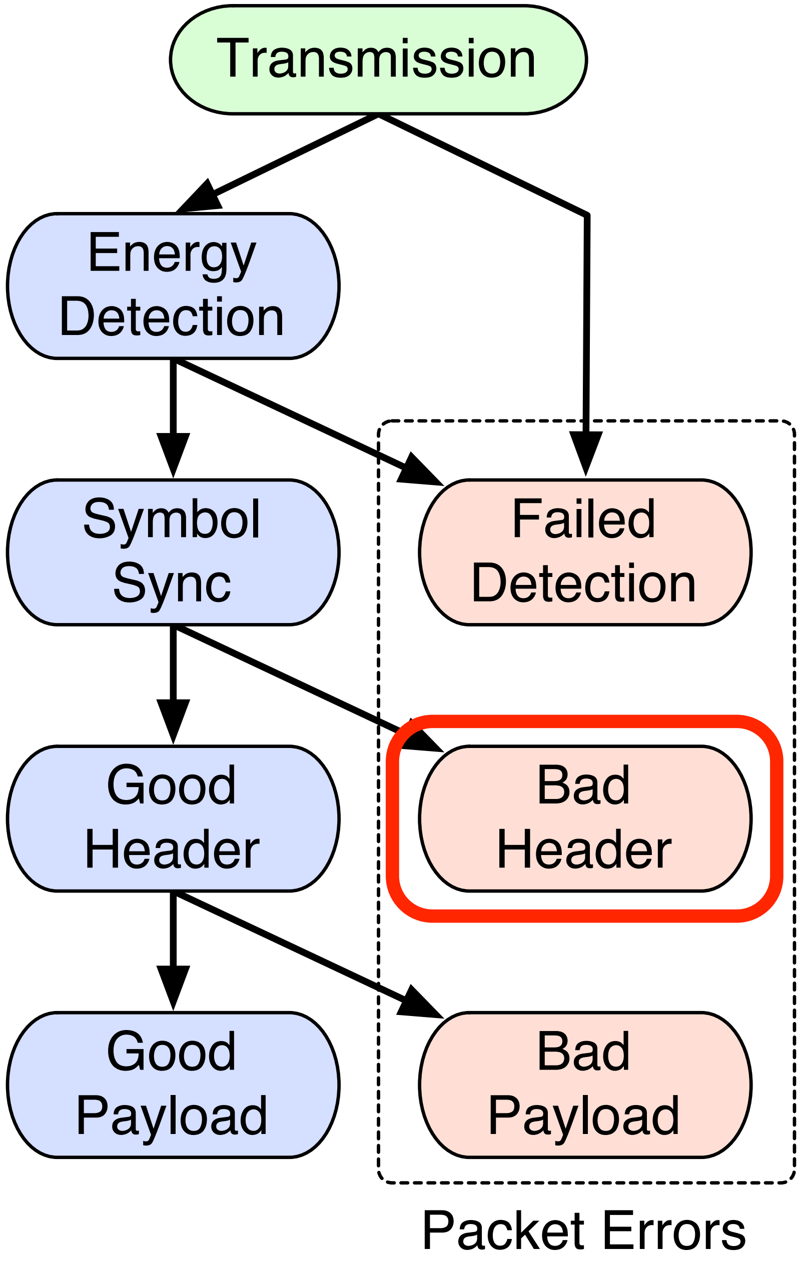


*QPSK /1412 bytes*

*Flat fading*

*6.9M packets*





Transmission

Energy Detection

Symbol Sync

Good Header

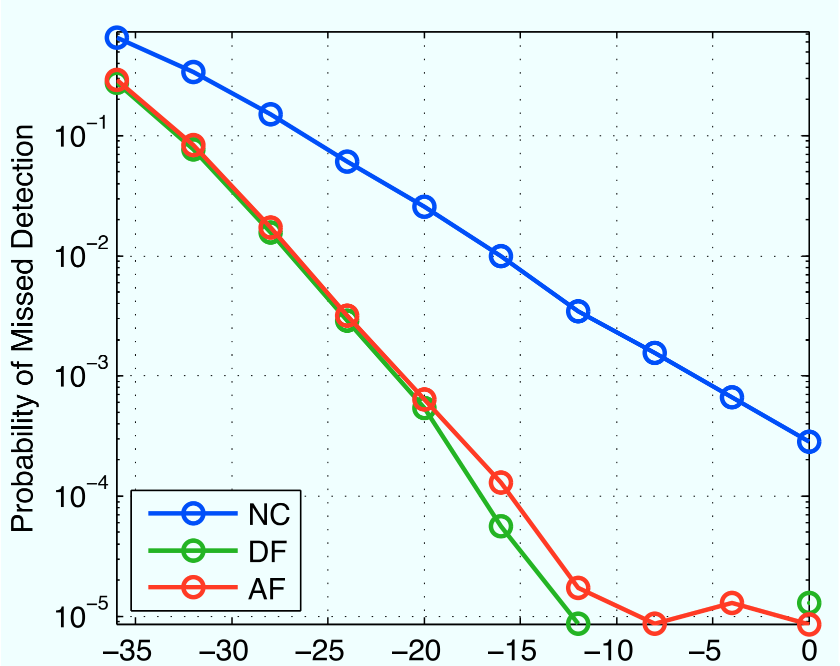
Good Payload

Failed Detection

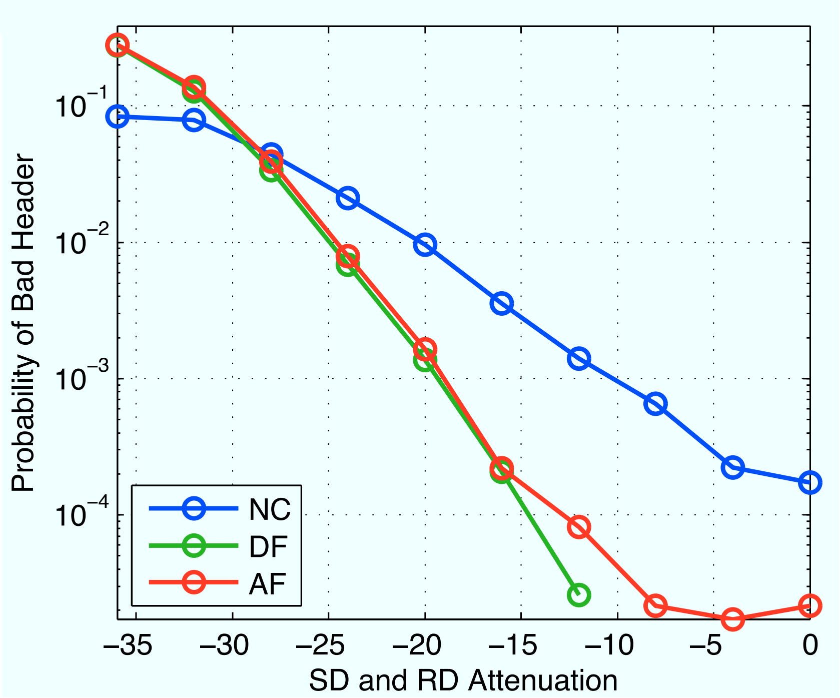
Bad Header

Bad Payload

Packet Errors



P(No Detection)

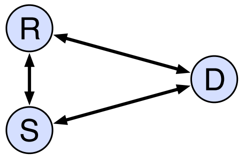


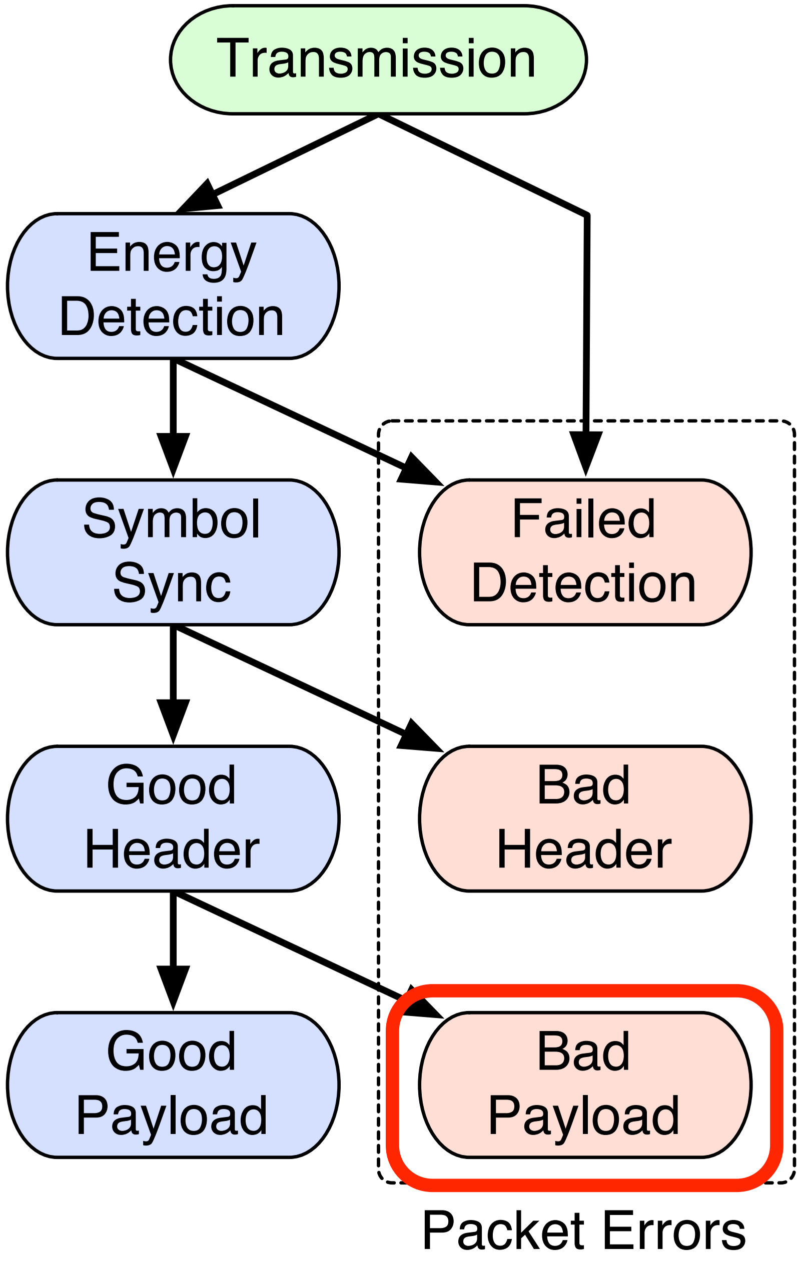
P(Bad Header)

*QPSK /1412 bytes*

*Flat fading*

*6.9M packets*





Transmission

Energy Detection

Symbol Sync

Good Header

Good Payload

Failed Detection

Bad Header

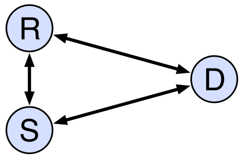
Bad Payload

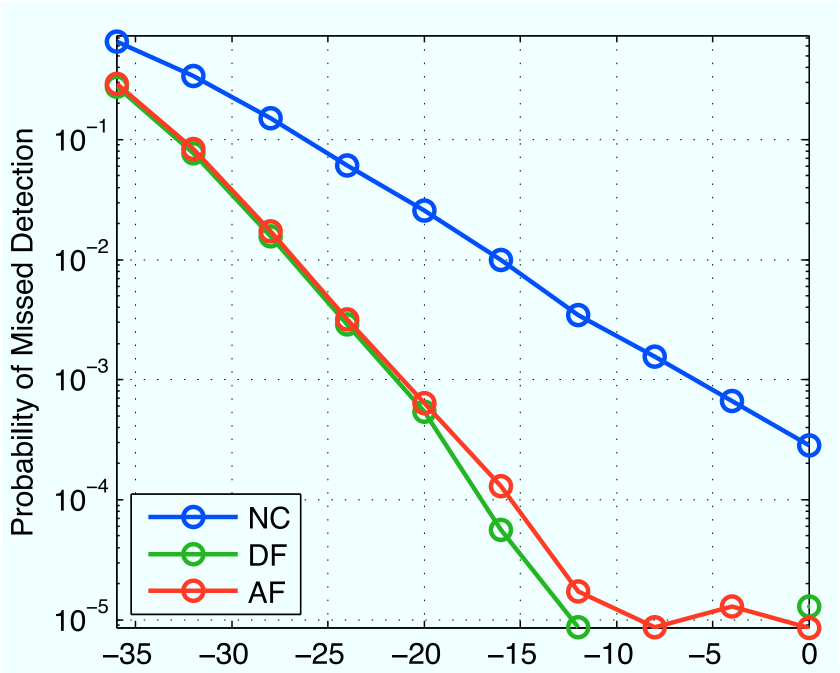
Packet Errors

*QPSK /1412 bytes*

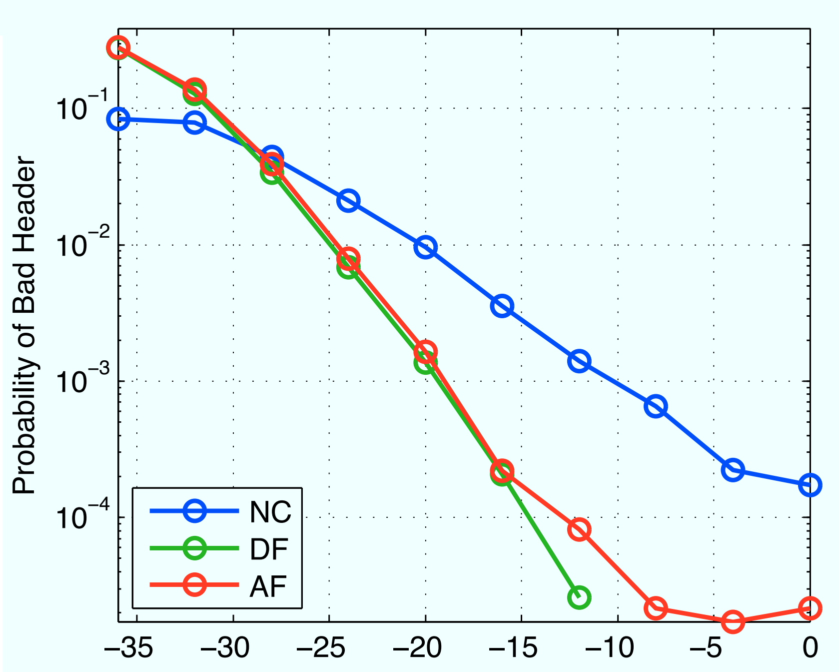
*Flat fading*

*6.9M packets*

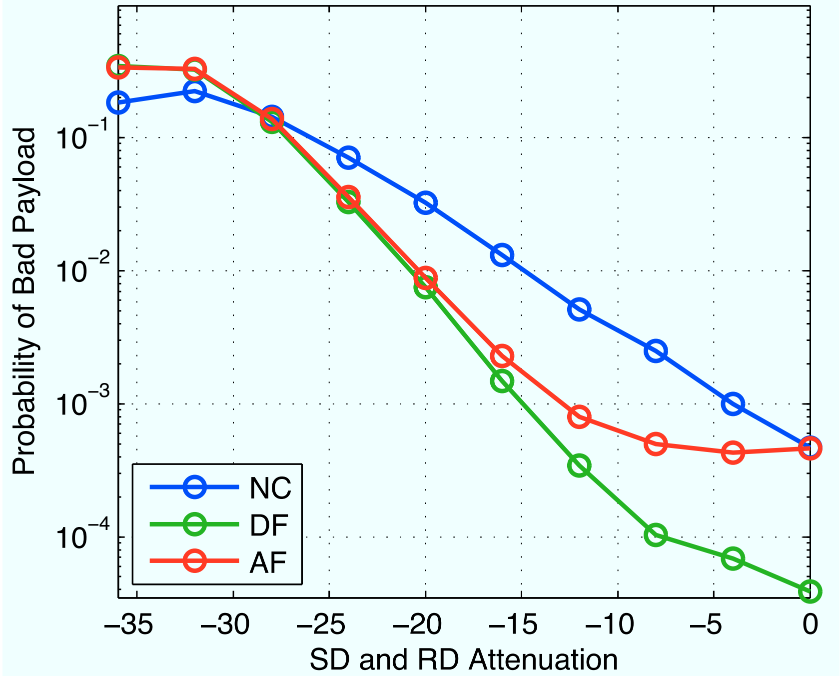




P(No Detection)

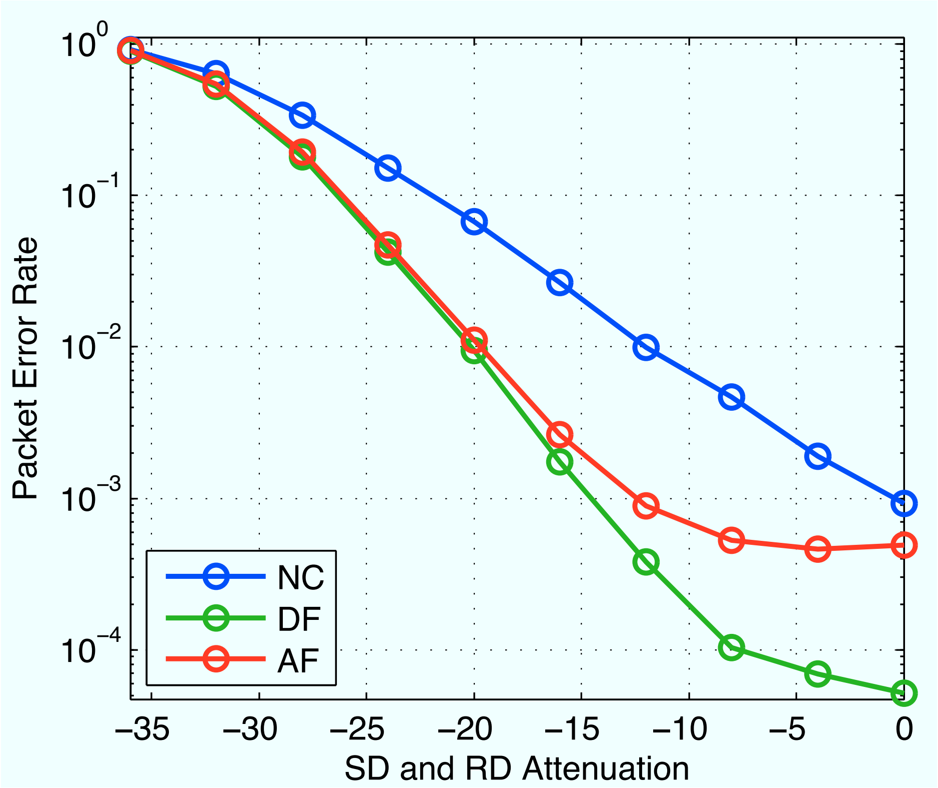


P(Bad Header)

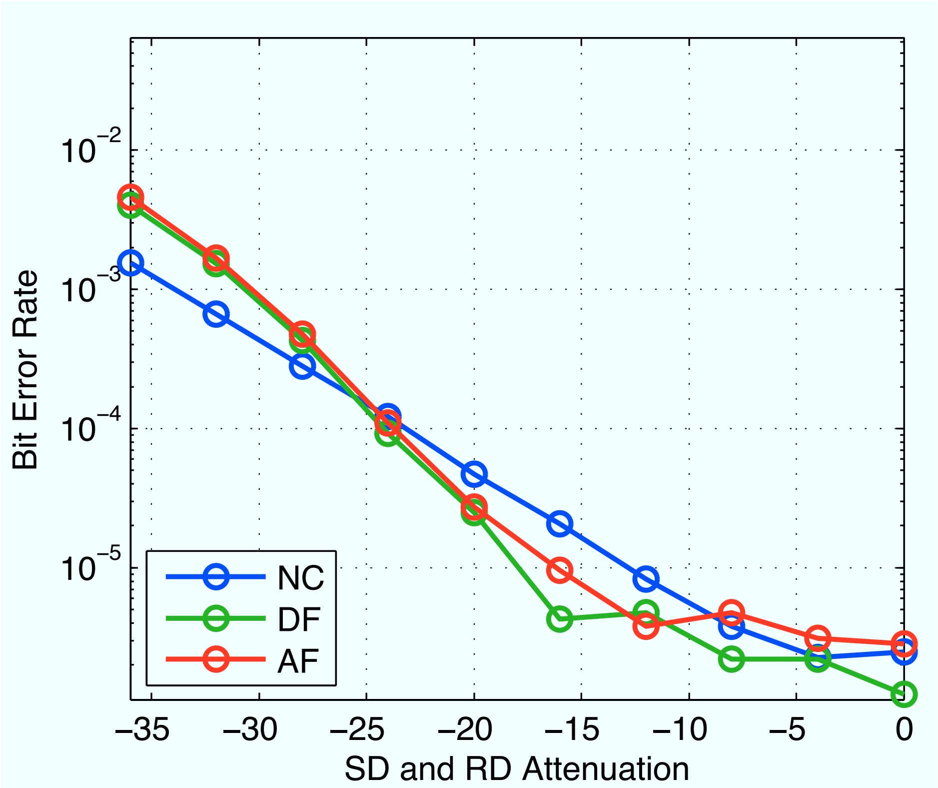


P(Bad Payload)

PER



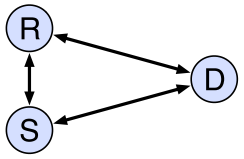
BER

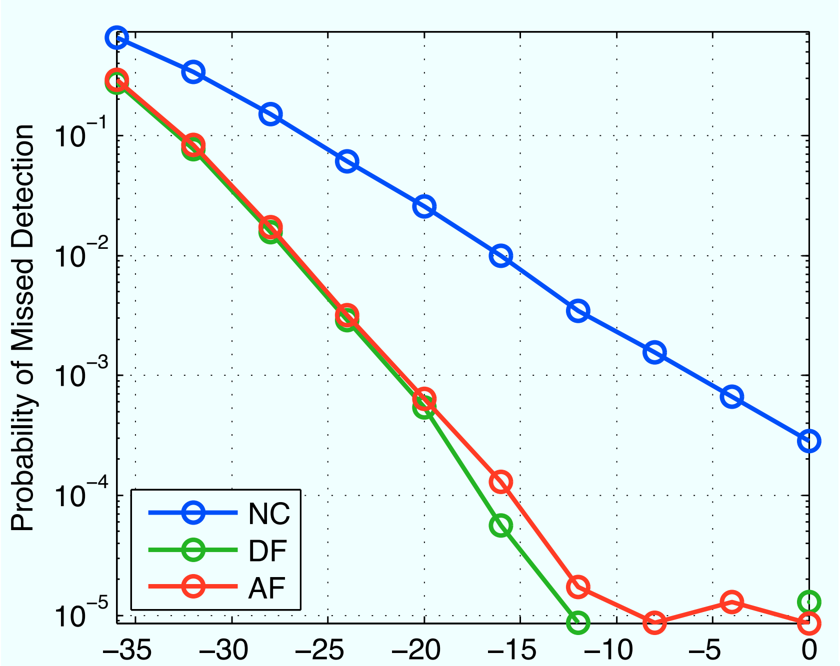


*QPSK /1412 bytes*

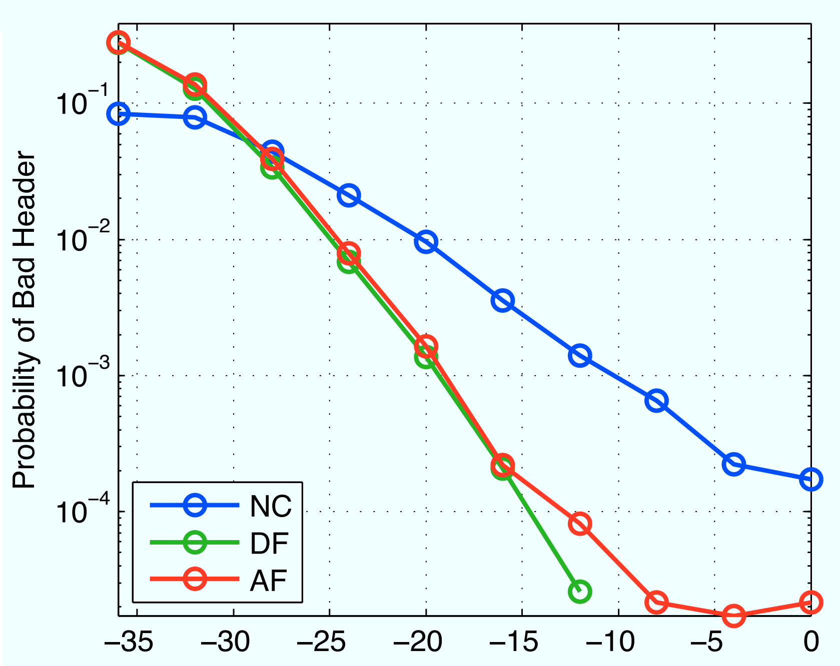
*Flat fading*

*6.9M packets*

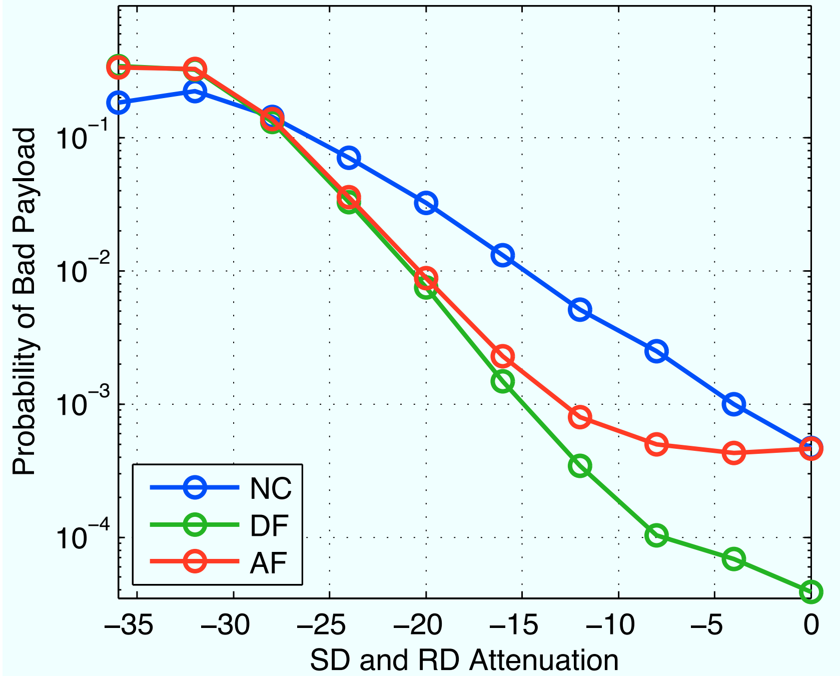




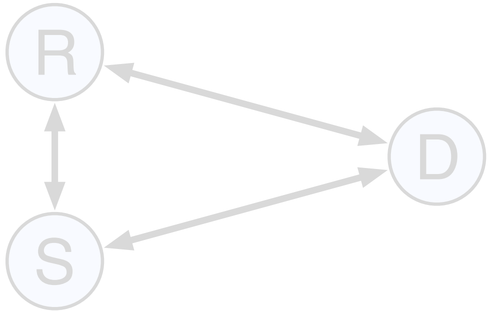
P(No Detection)

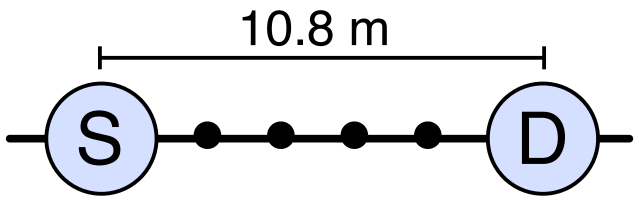


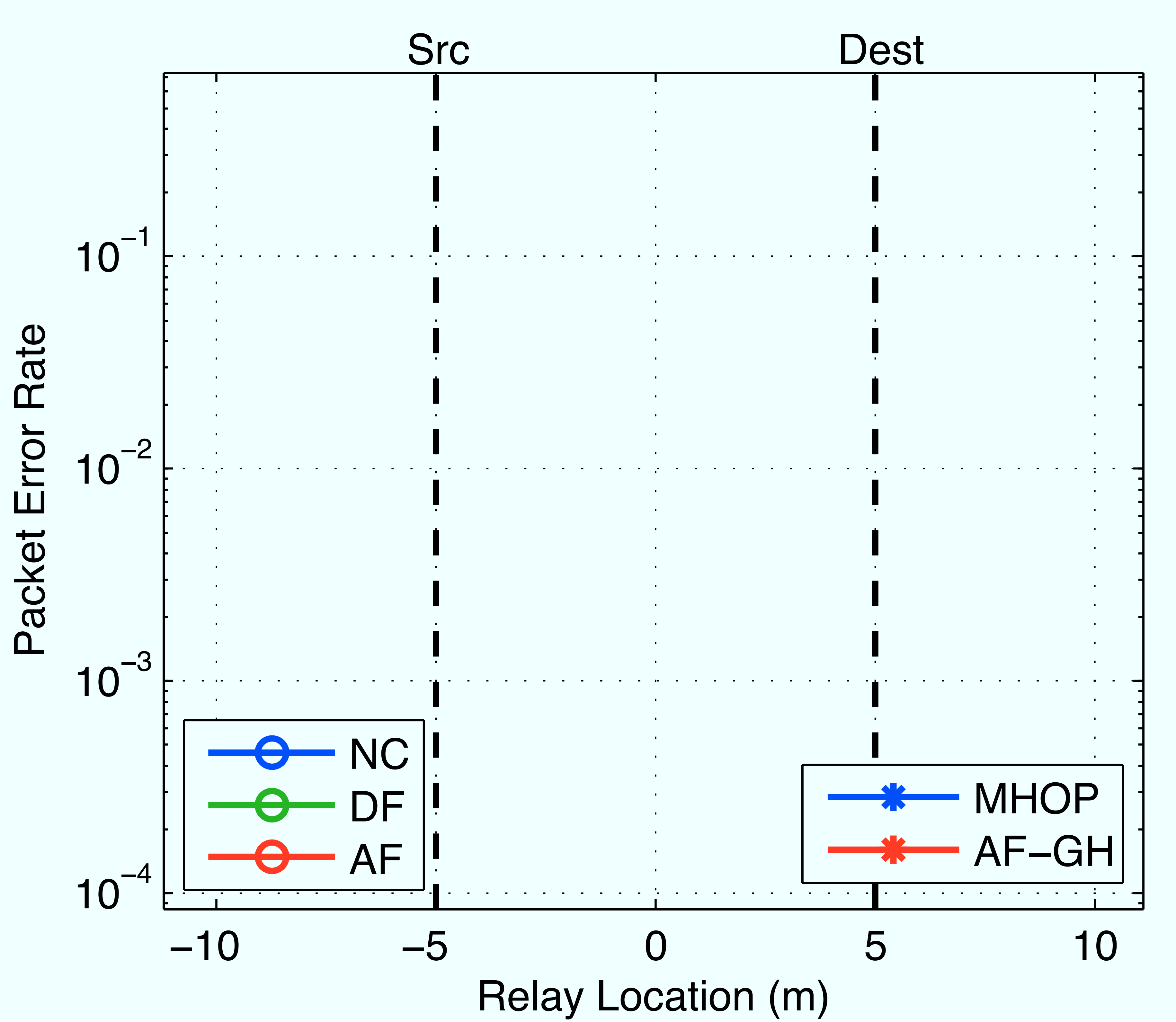
P(Bad Header)



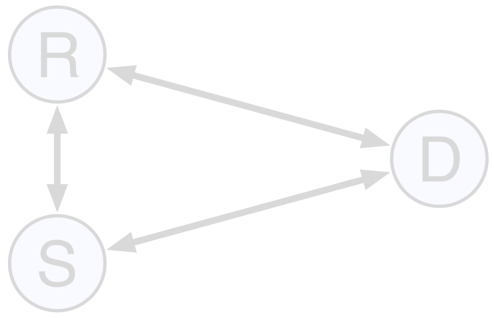
P(Bad Payload)

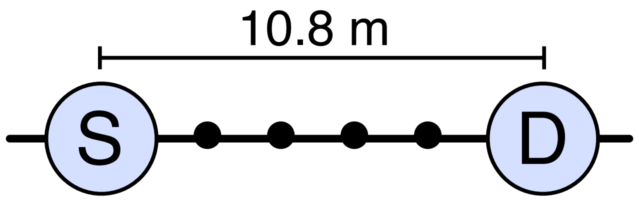


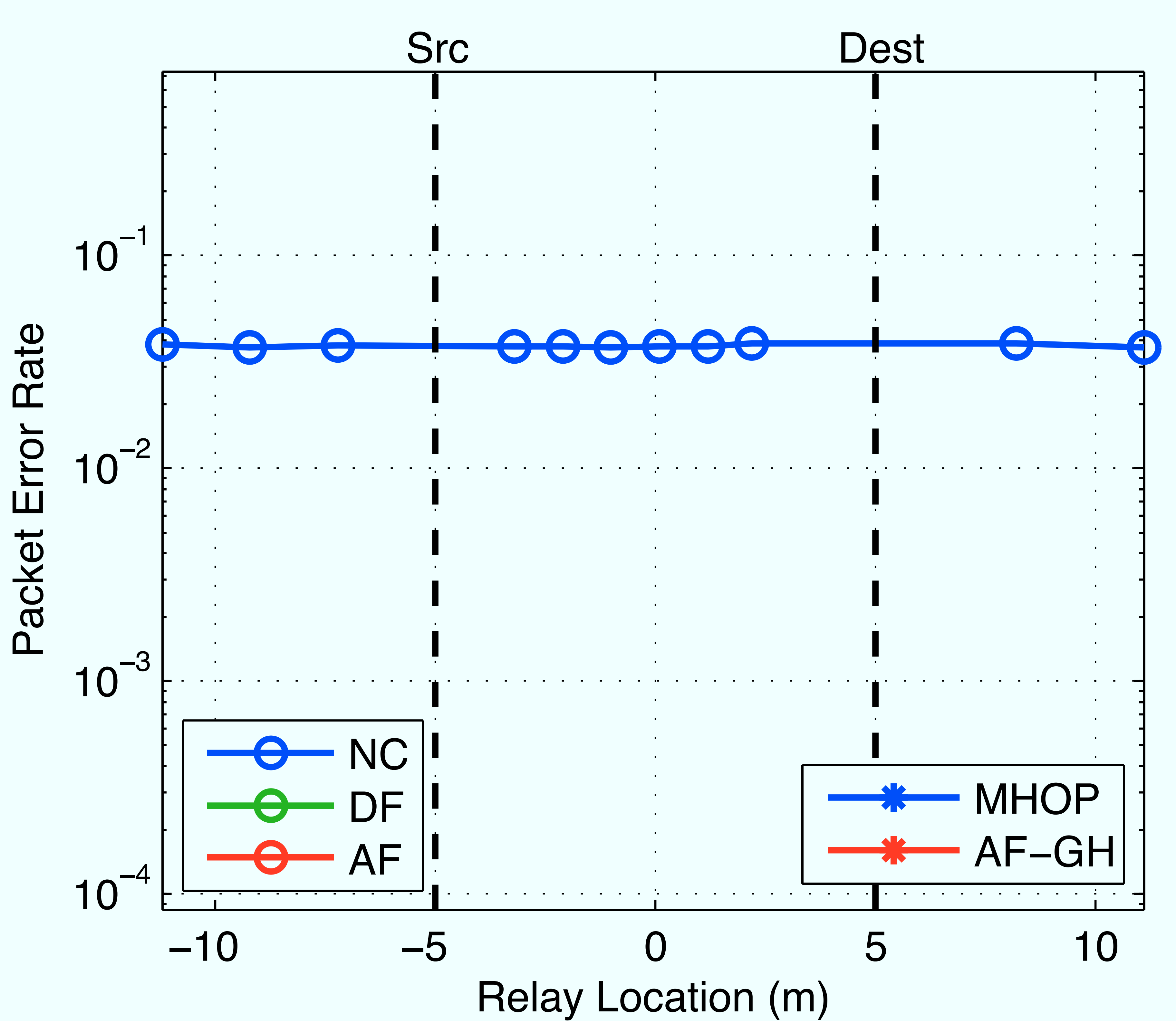




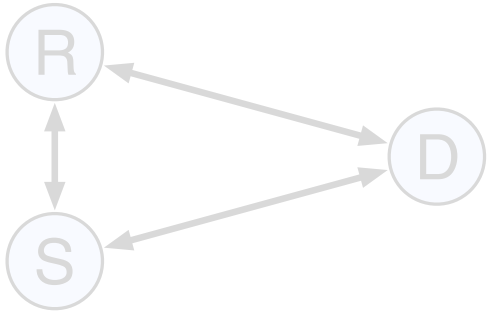
*QPSK*
*1412 bytes*
*17.6M packets*

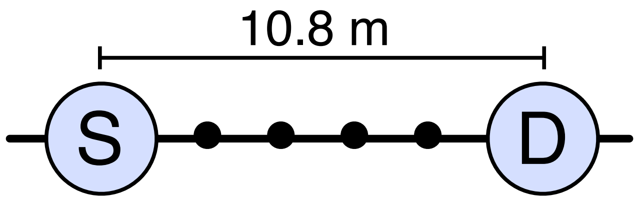


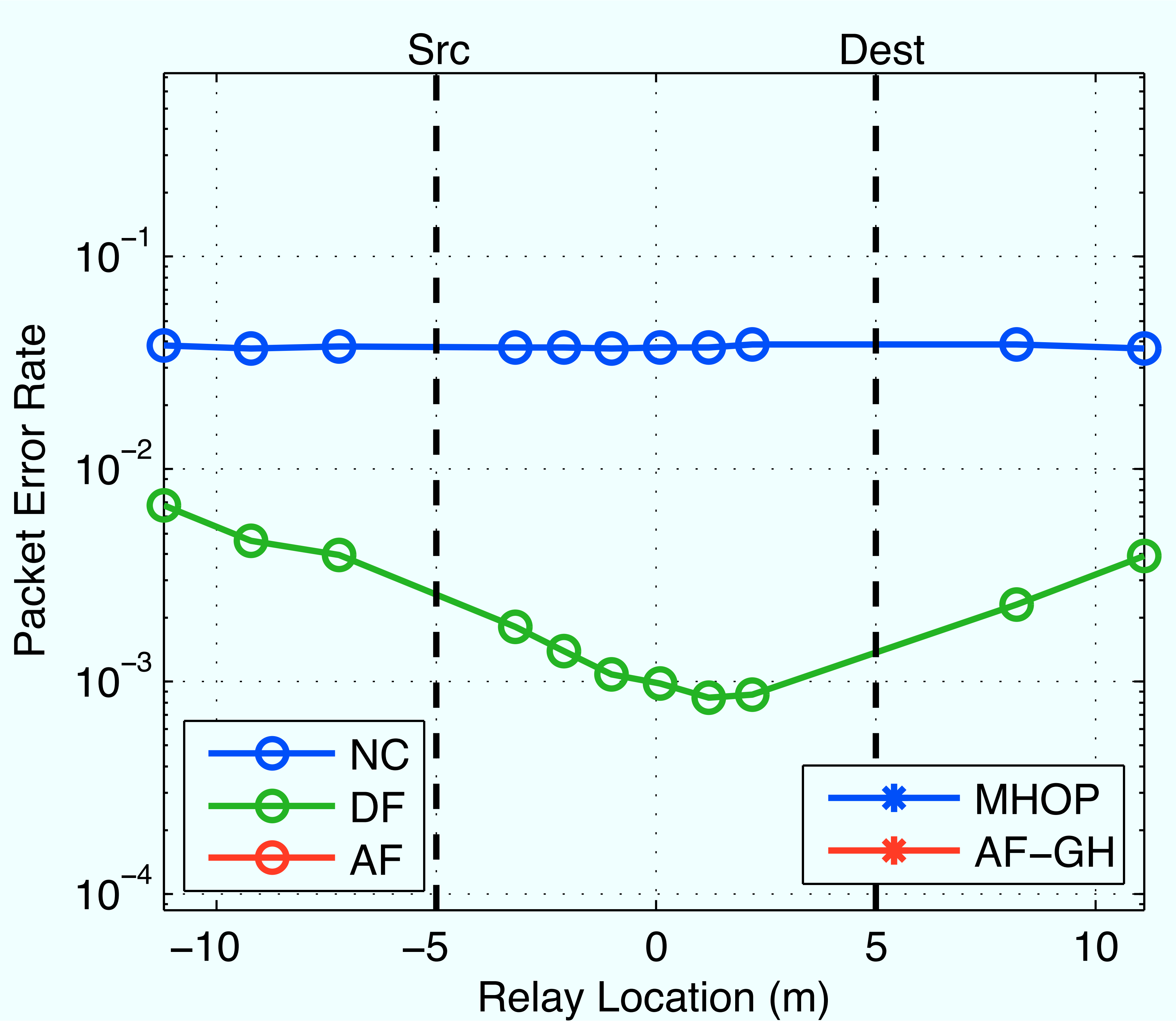




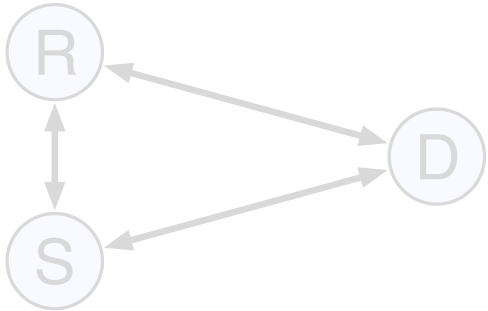
*QPSK*
*1412 bytes*
*17.6M packets*

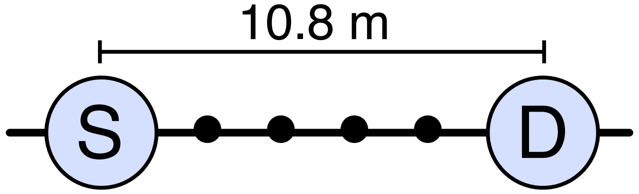


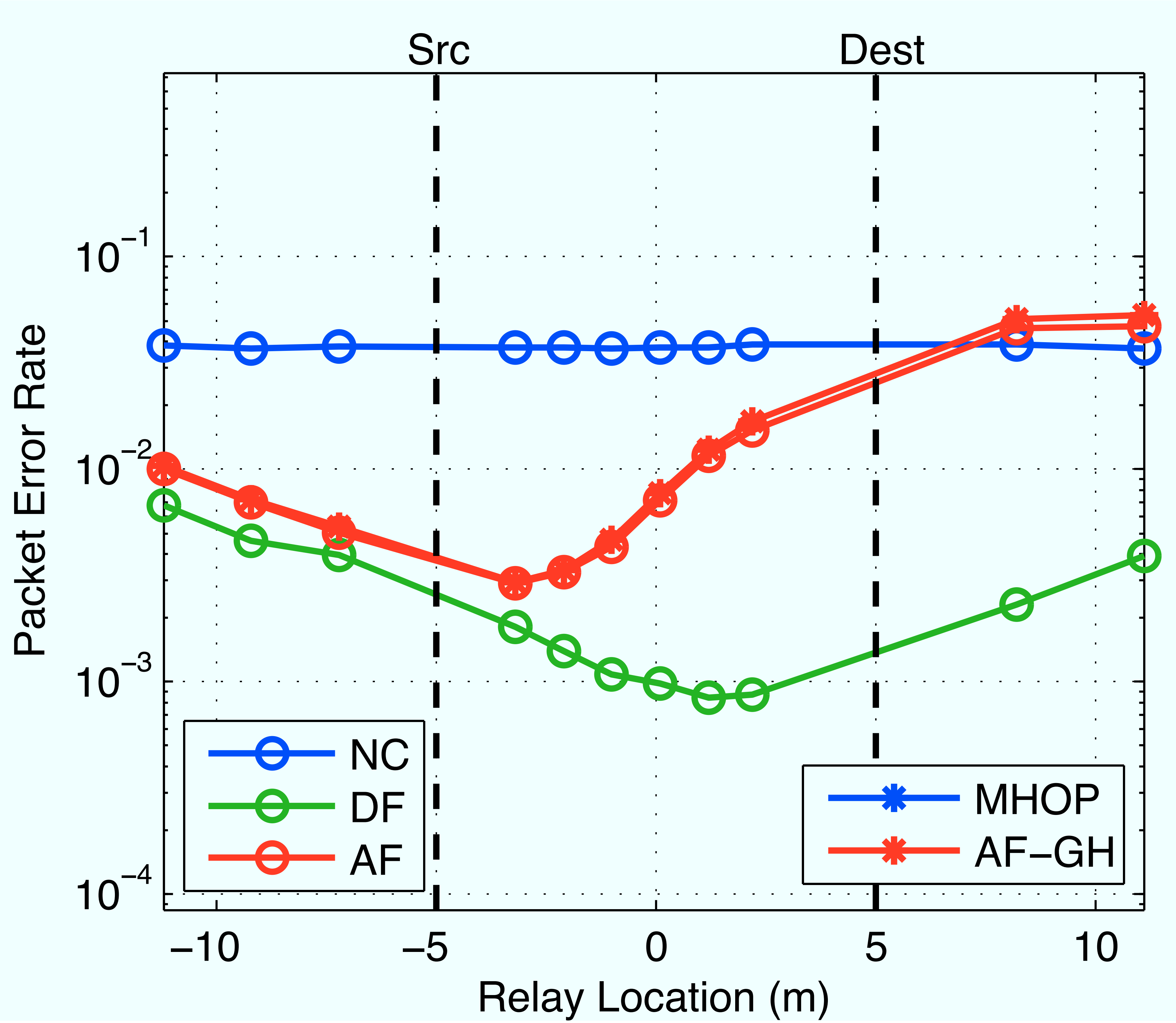




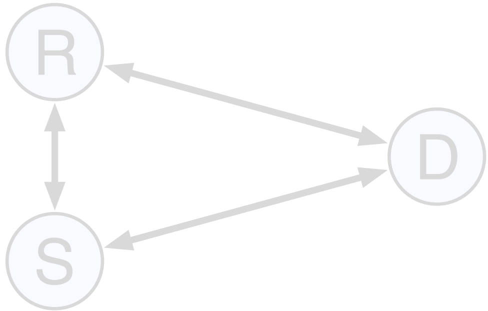
*QPSK*
*1412 bytes*
*17.6M packets*

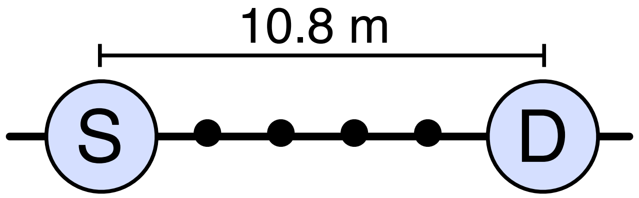


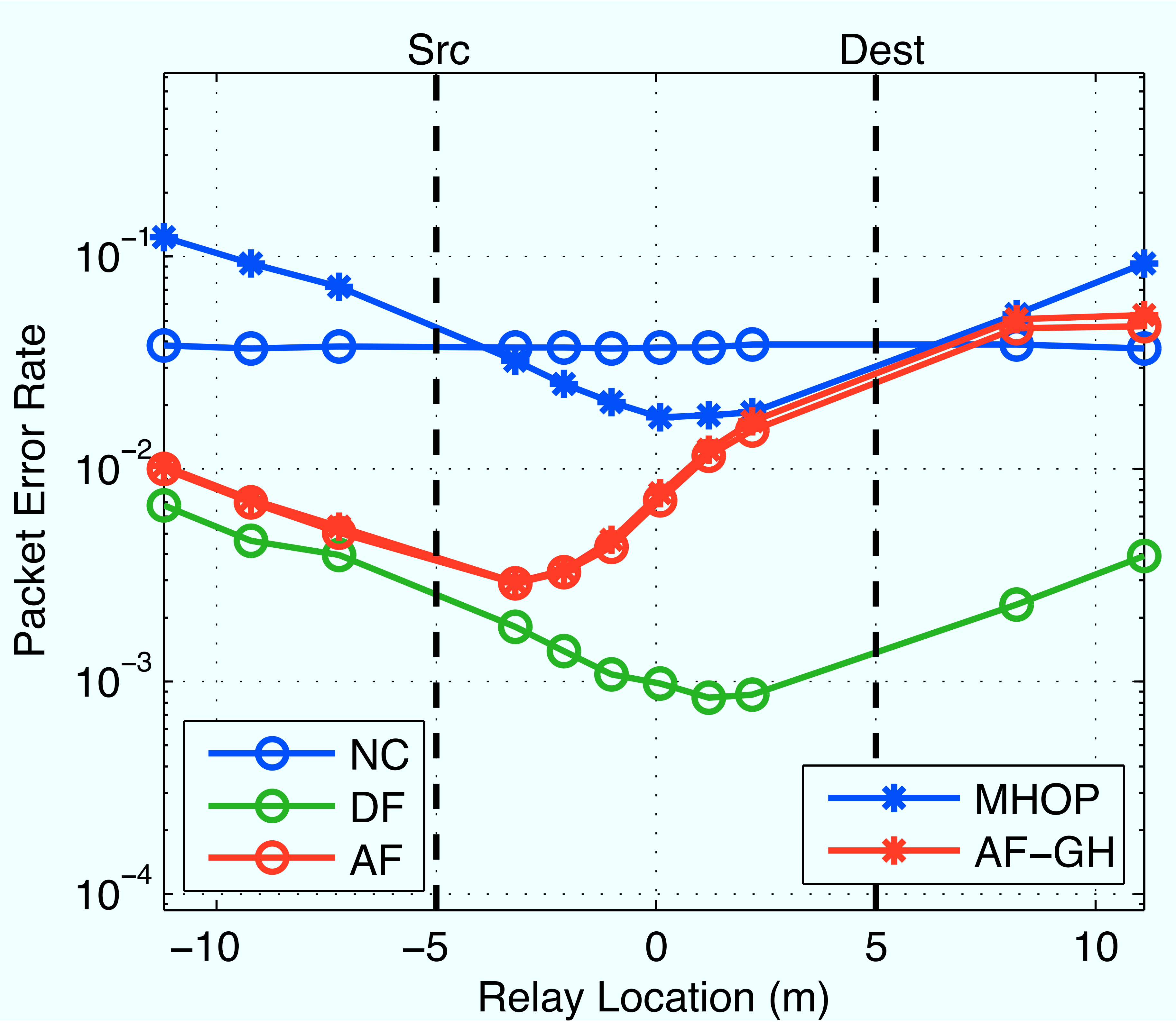




*QPSK*
*1412 bytes*
*17.6M packets*



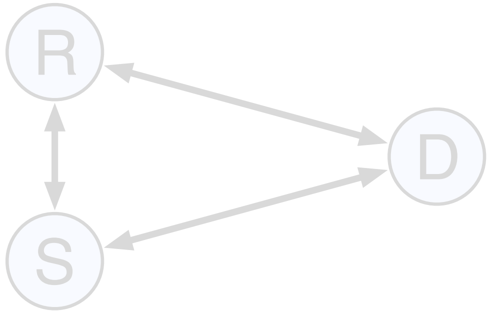


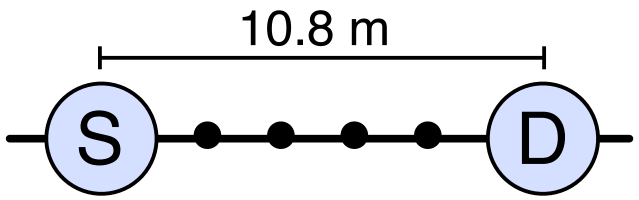


*QPSK*

*1412 bytes*

*17.6M packets*



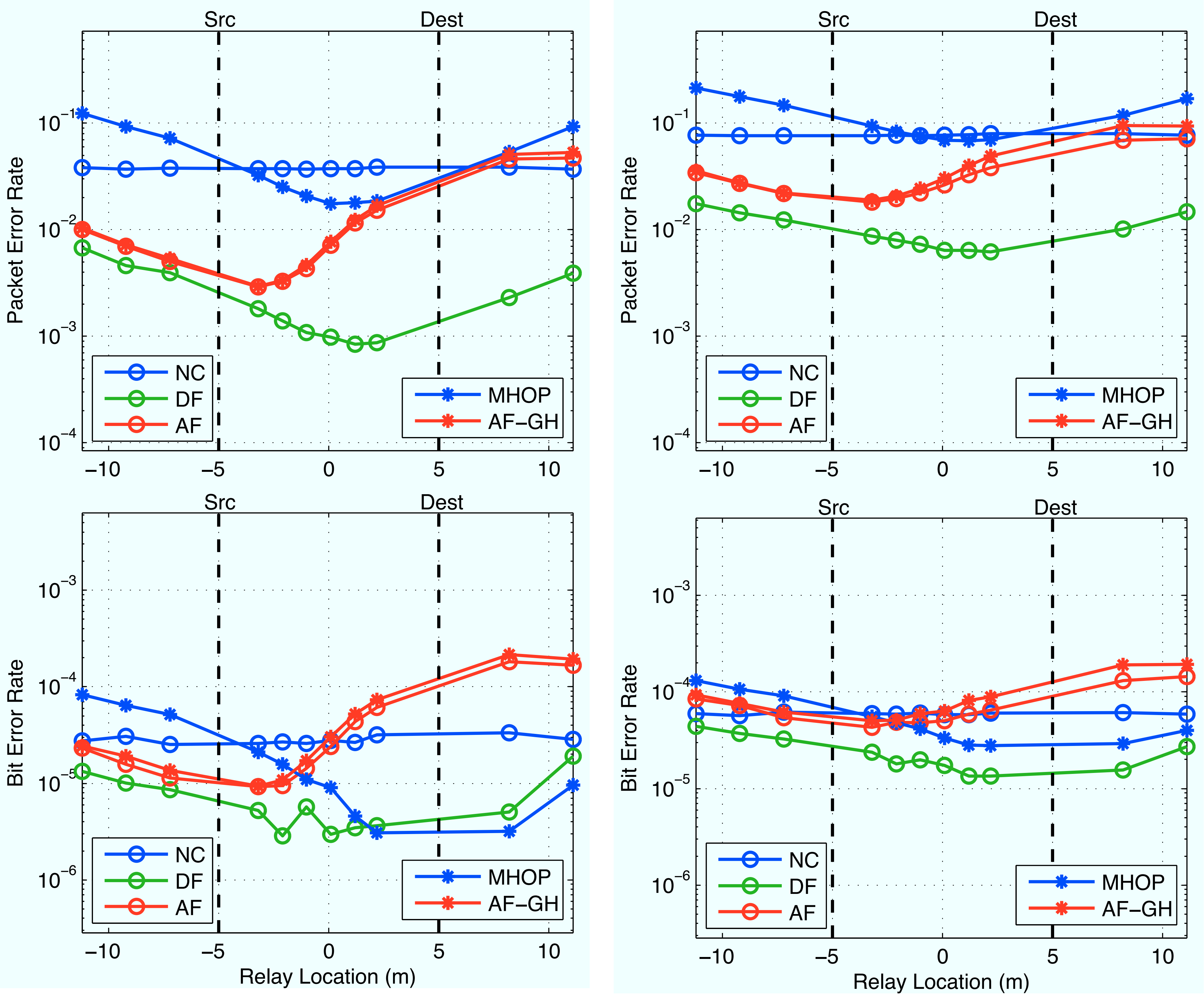


Frequency Flat

Frequency Selective

PER

BER

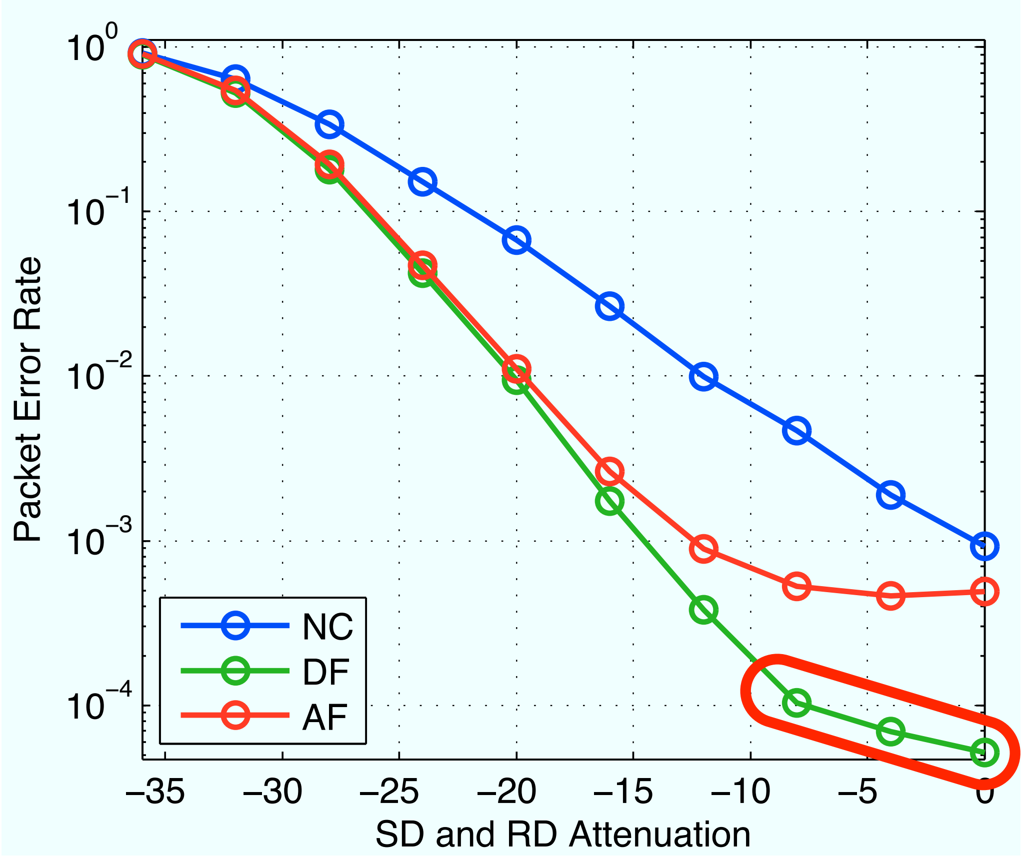


*QPSK*

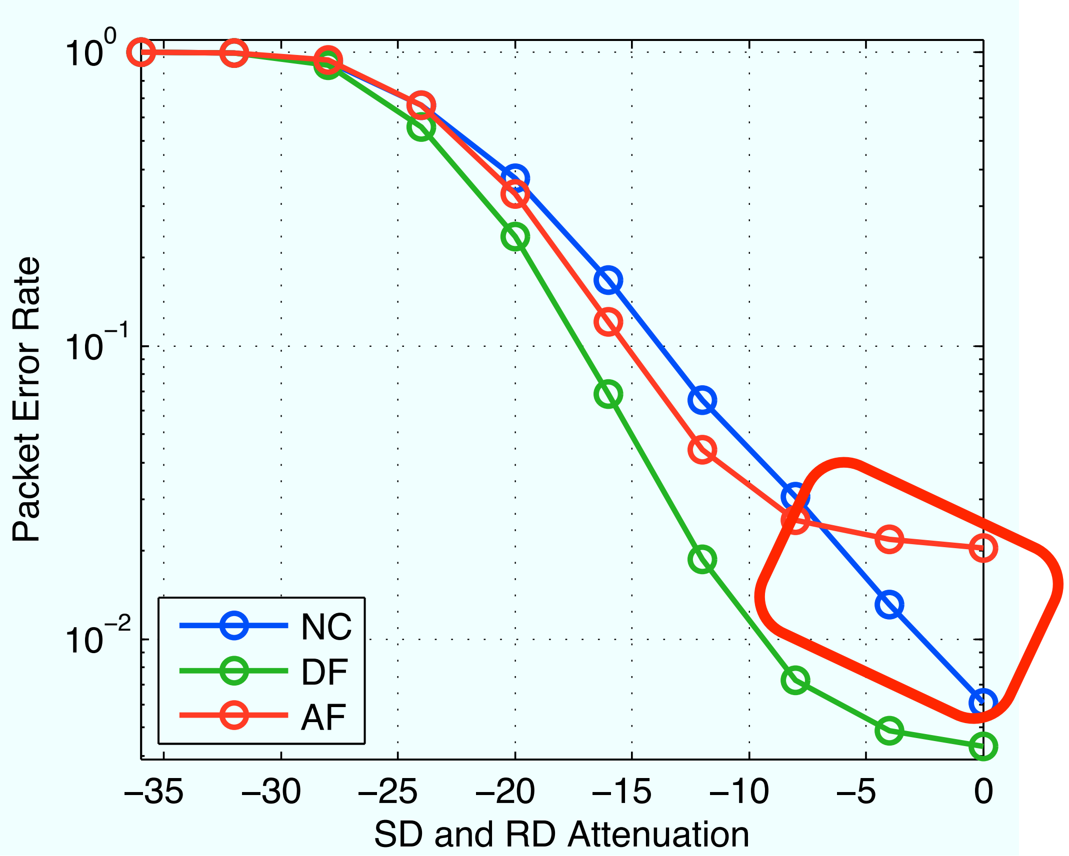
*1412 bytes*

*17.6M packets*

# Understanding the Results

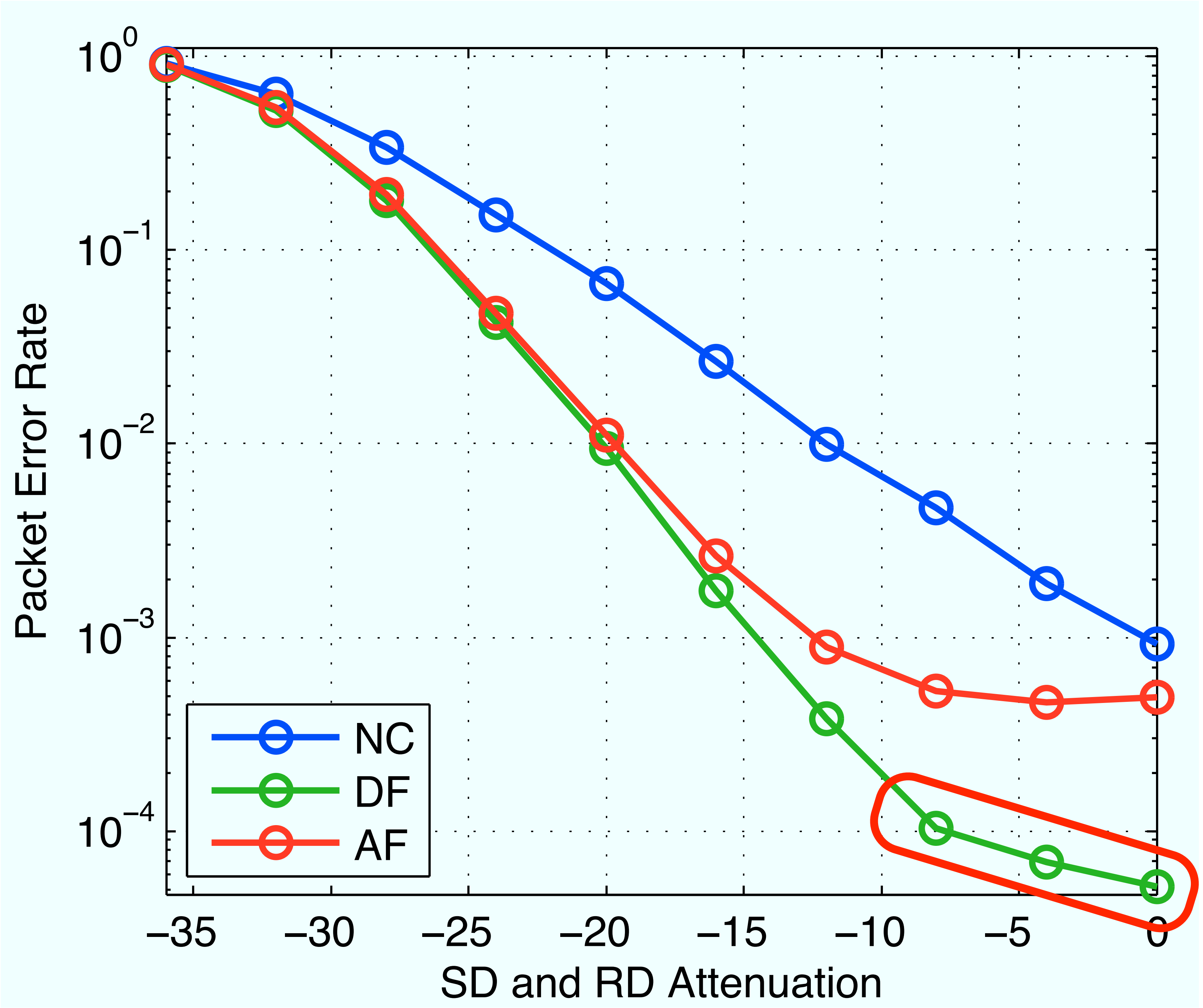


DF Floor



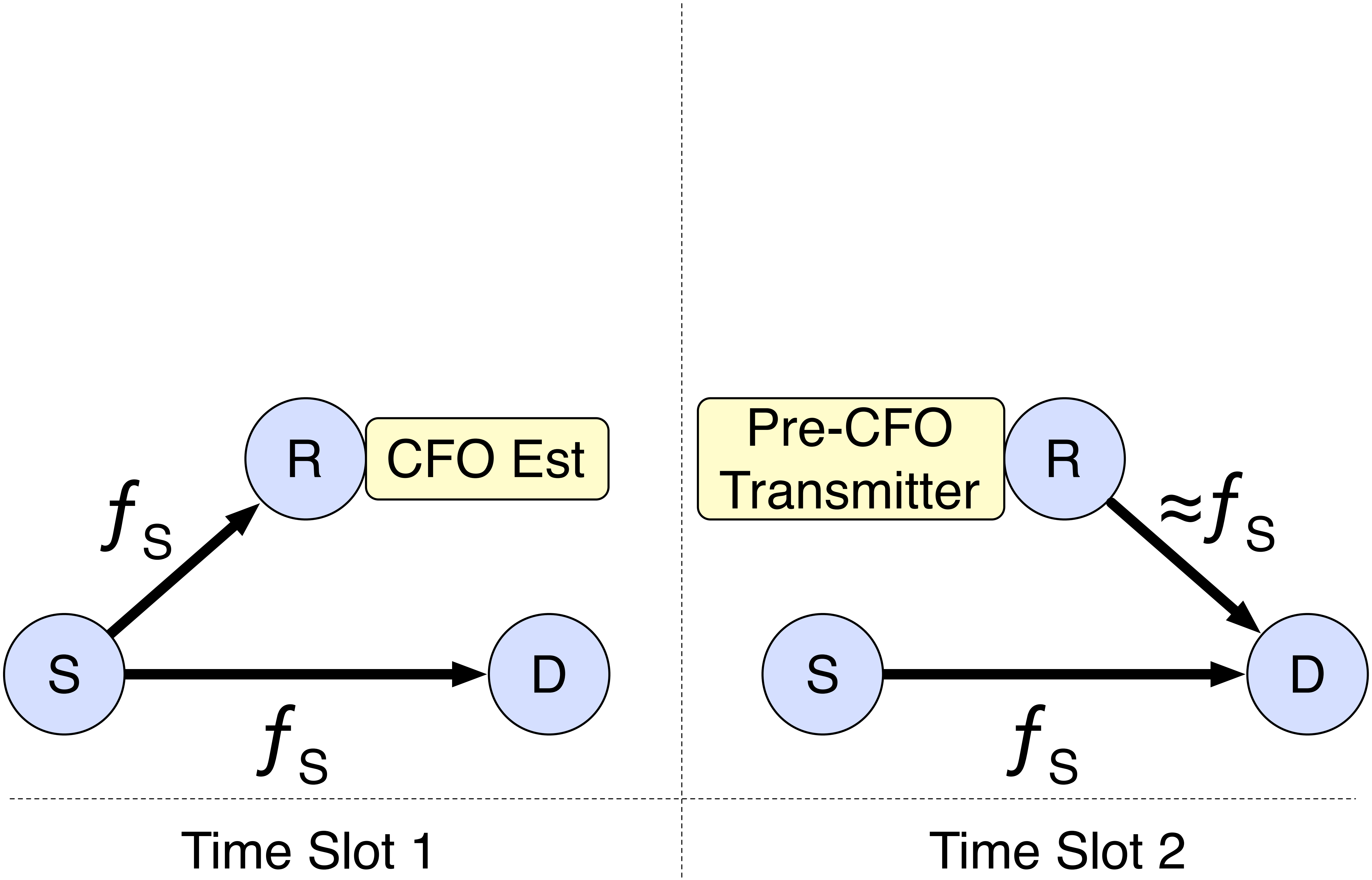
AF Floor

# Relay CFO Errors



- DF floor
- Occasional CFO estimation errors?

# Relay CFO Errors



R

CFO Est

$f_S$

S

D

$f_S$

Time Slot 1

Pre-CFO Transmitter

R

$\approx f_S$

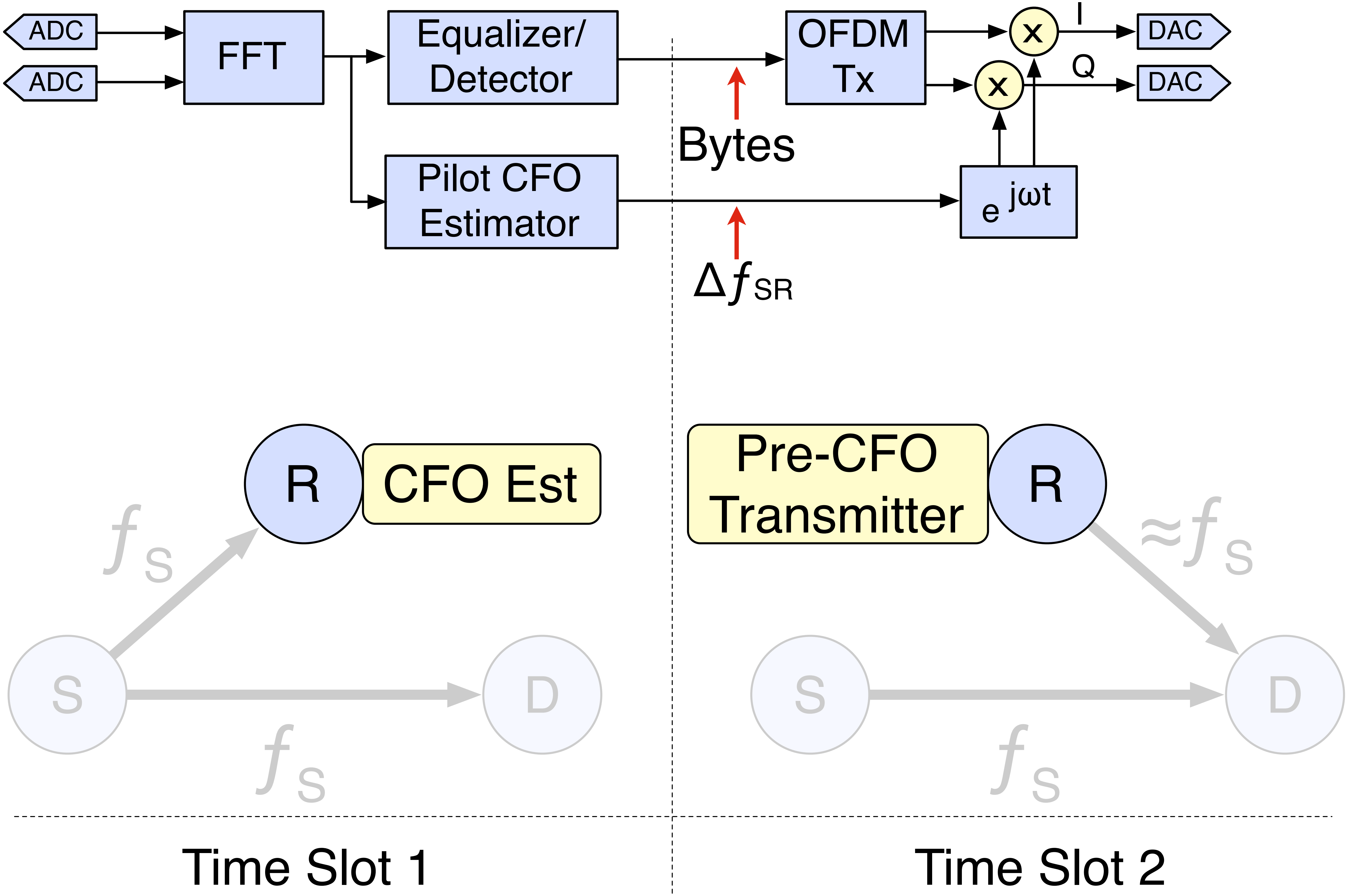
S

D

$f_S$

Time Slot 2

# Relay CFO Errors



ADC
ADC
FFT
Equalizer/ Detector
Pilot CFO Estimator
OFDM Tx
$e^{j\omega t}$
x
x
I
Q
DAC
DAC
Bytes
$\Delta f_{SR}$

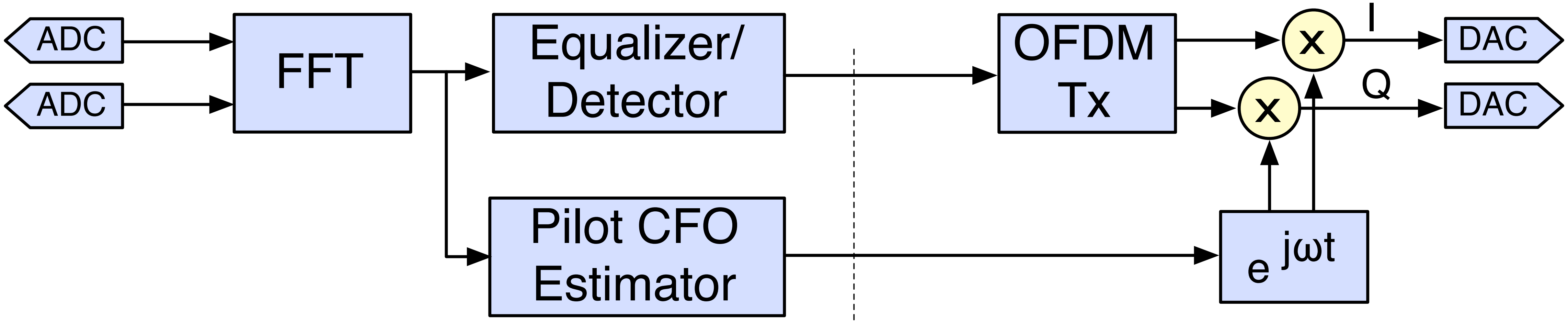
R
CFO Est
S
D
$f_S$
$f_S$

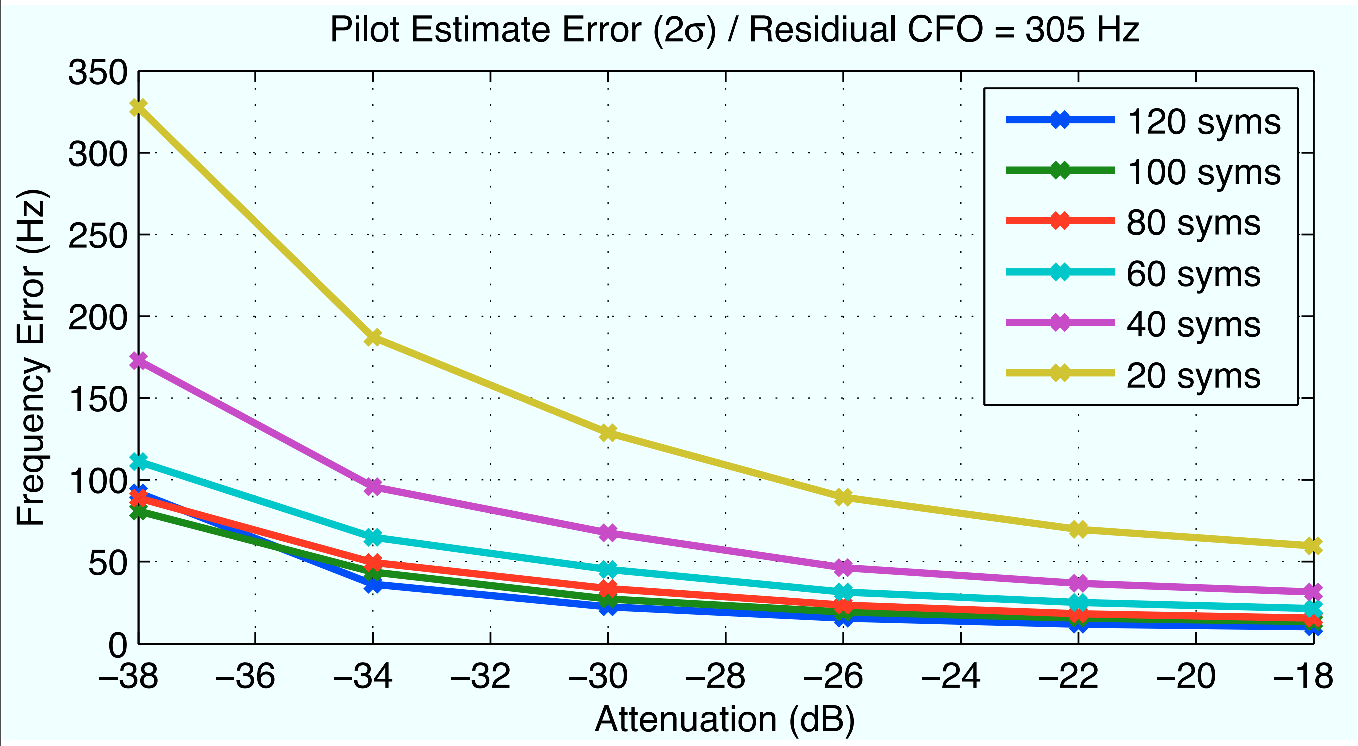
Pre-CFO Transmitter
R
S
D
$\approx f_S$
$f_S$

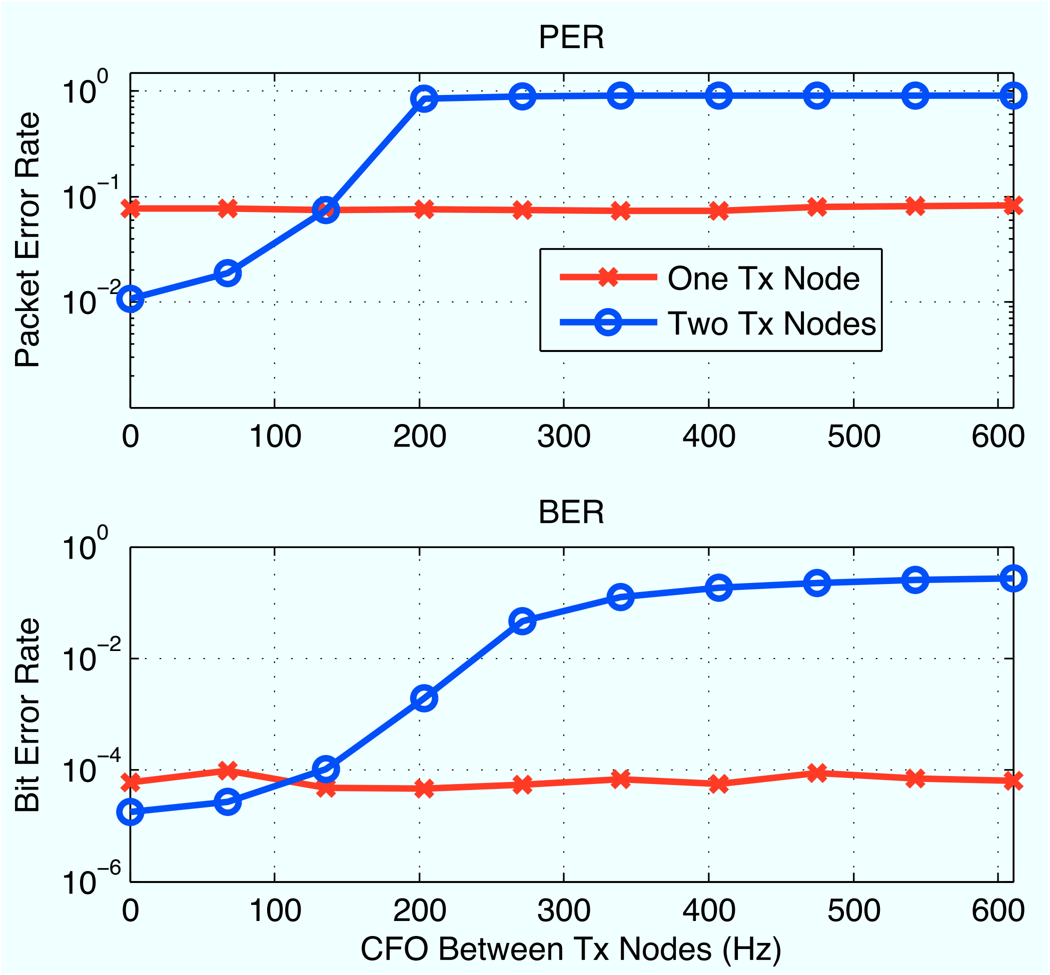
Time Slot 1

Time Slot 2

# Relay CFO Errors



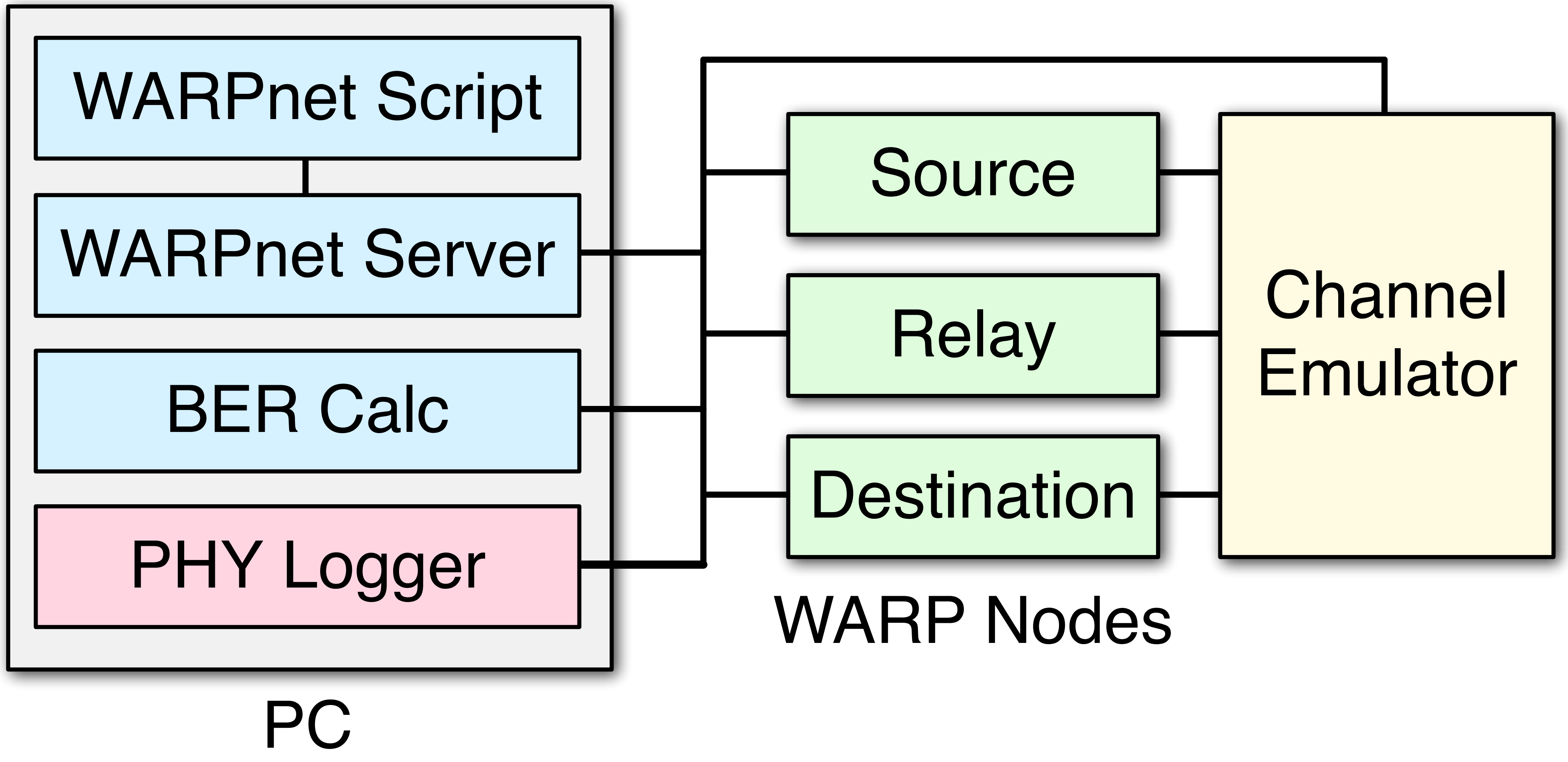




Time Slot 1

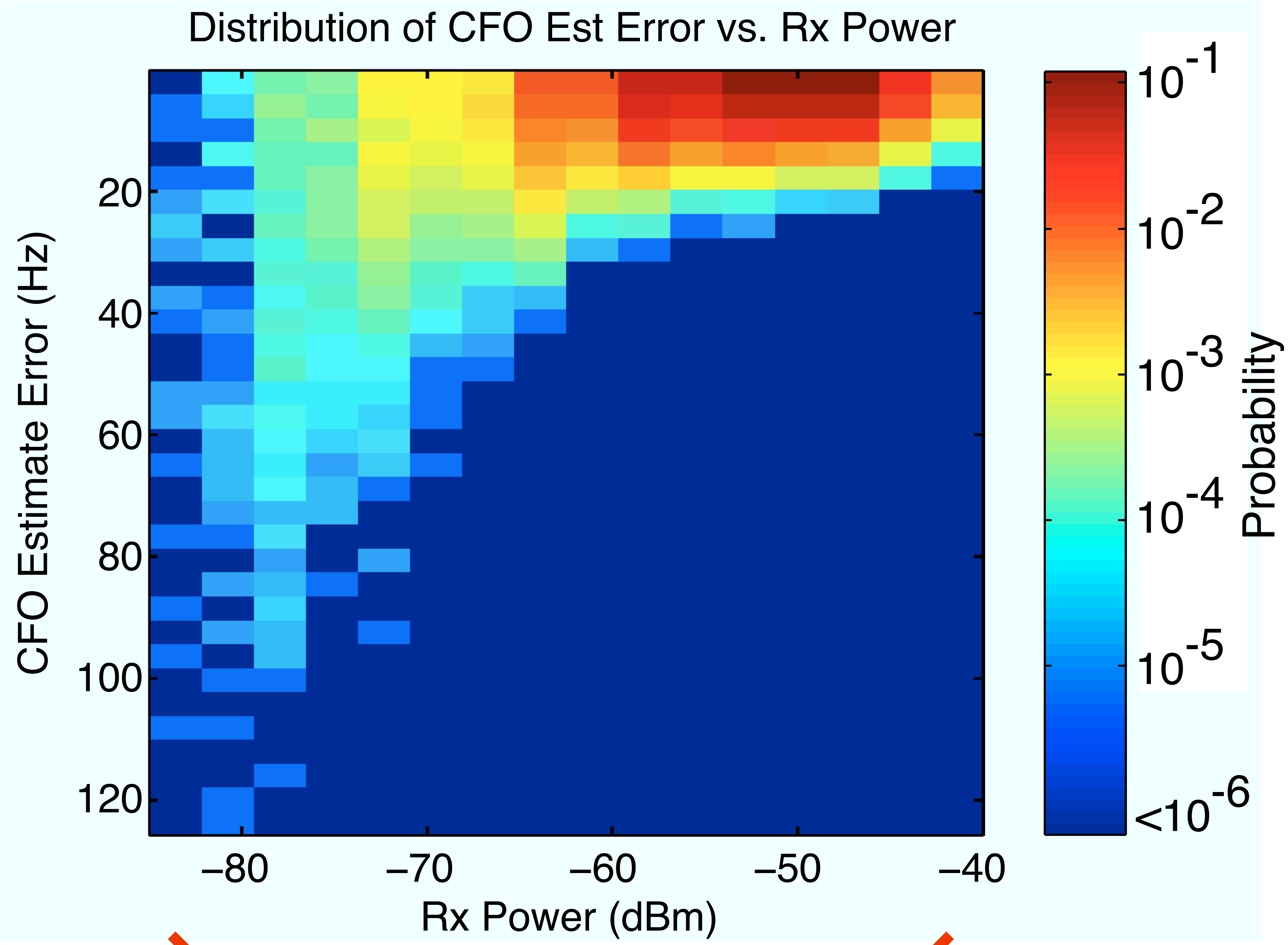
Time Slot 2

# WARPnet Framework



- PHY Logger co-processor acts like node
- Logs per-packet CFO estimates, AGC gains, EVM, RSSI, CRC status & channel coefficients

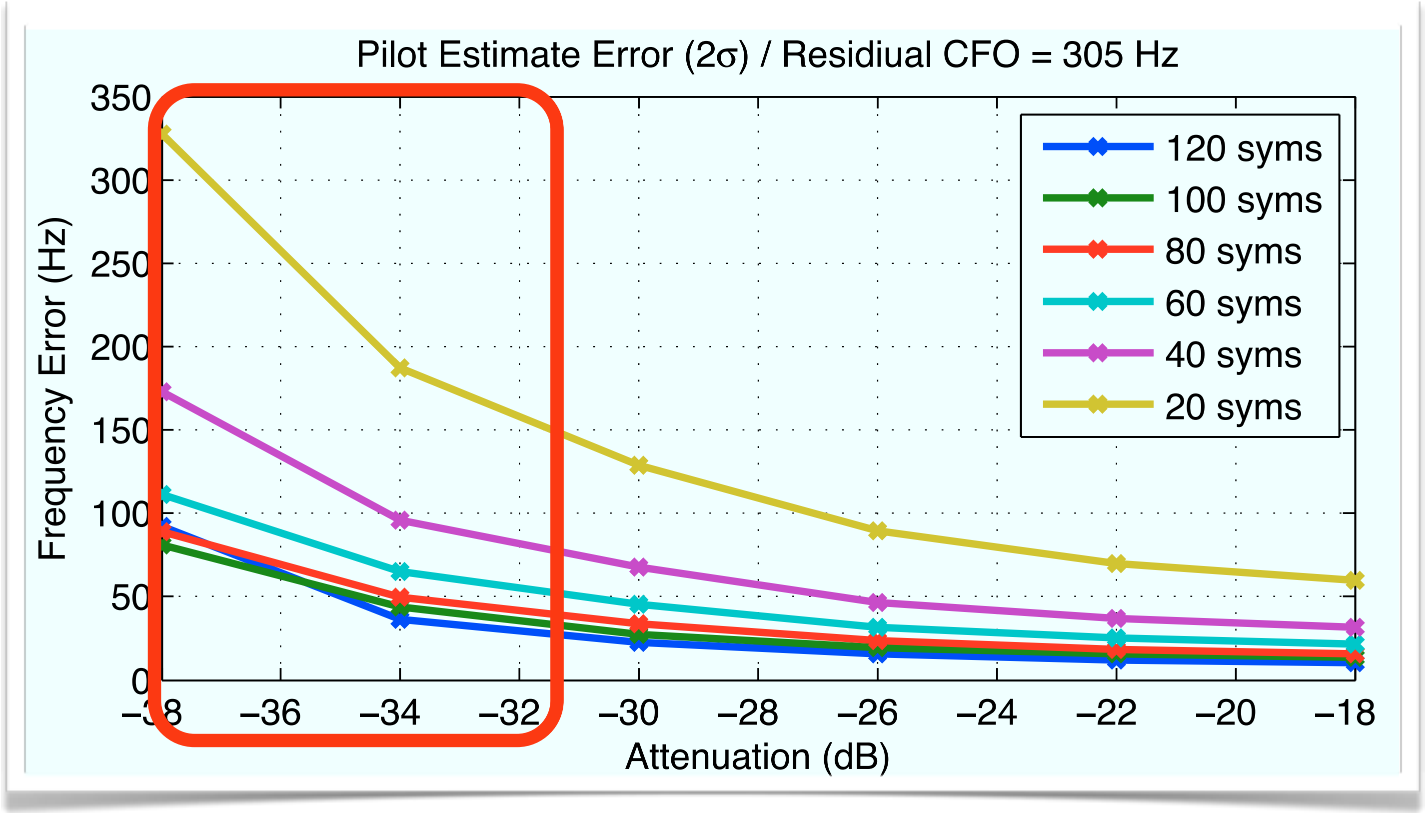
# Relay CFO Errors

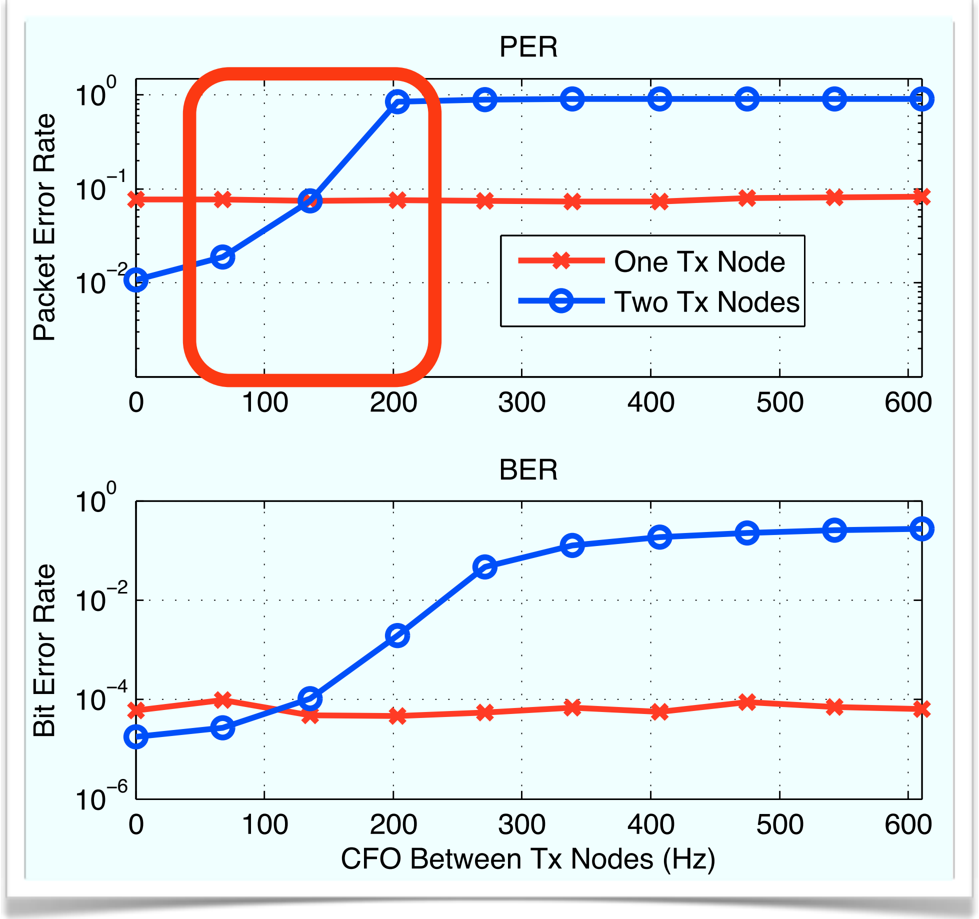


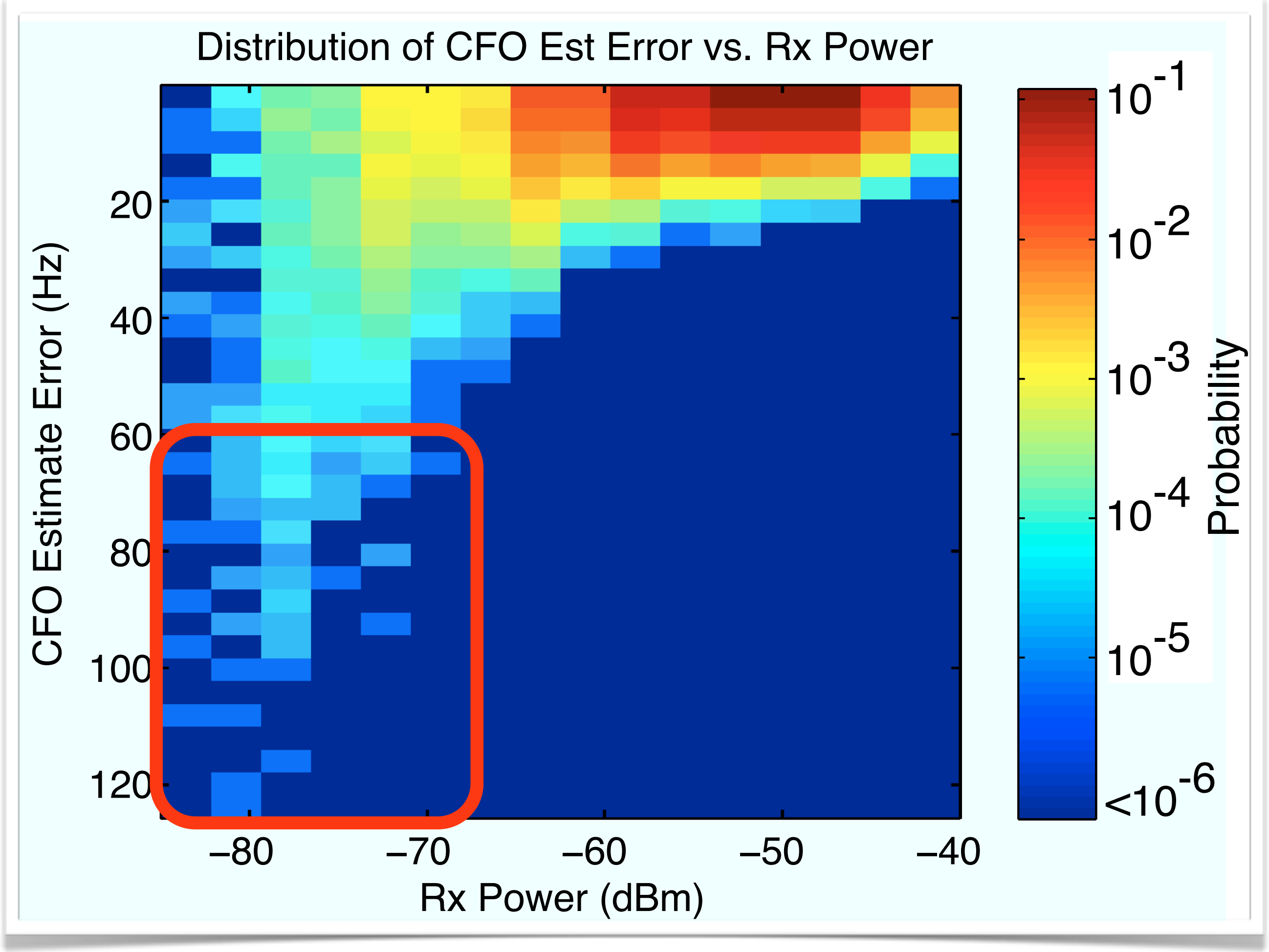
↑ Smaller Estimation Error

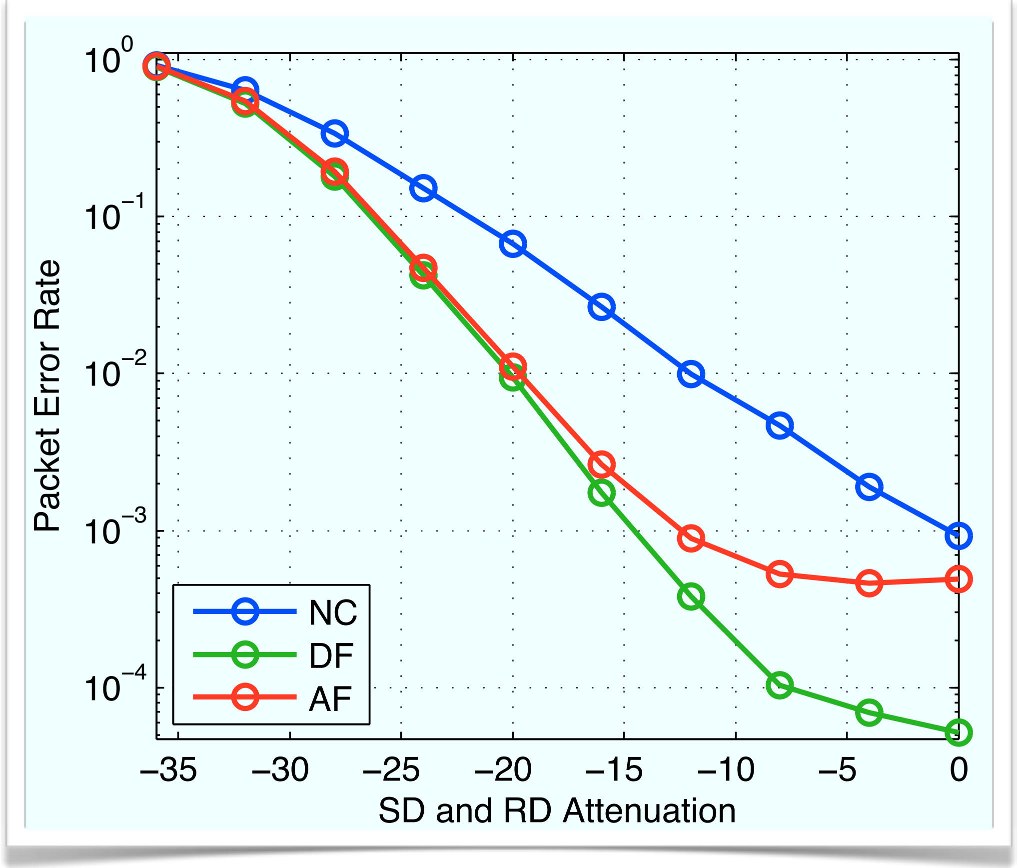
Fading Gives Rx Power Spread

# Relay CFO Errors

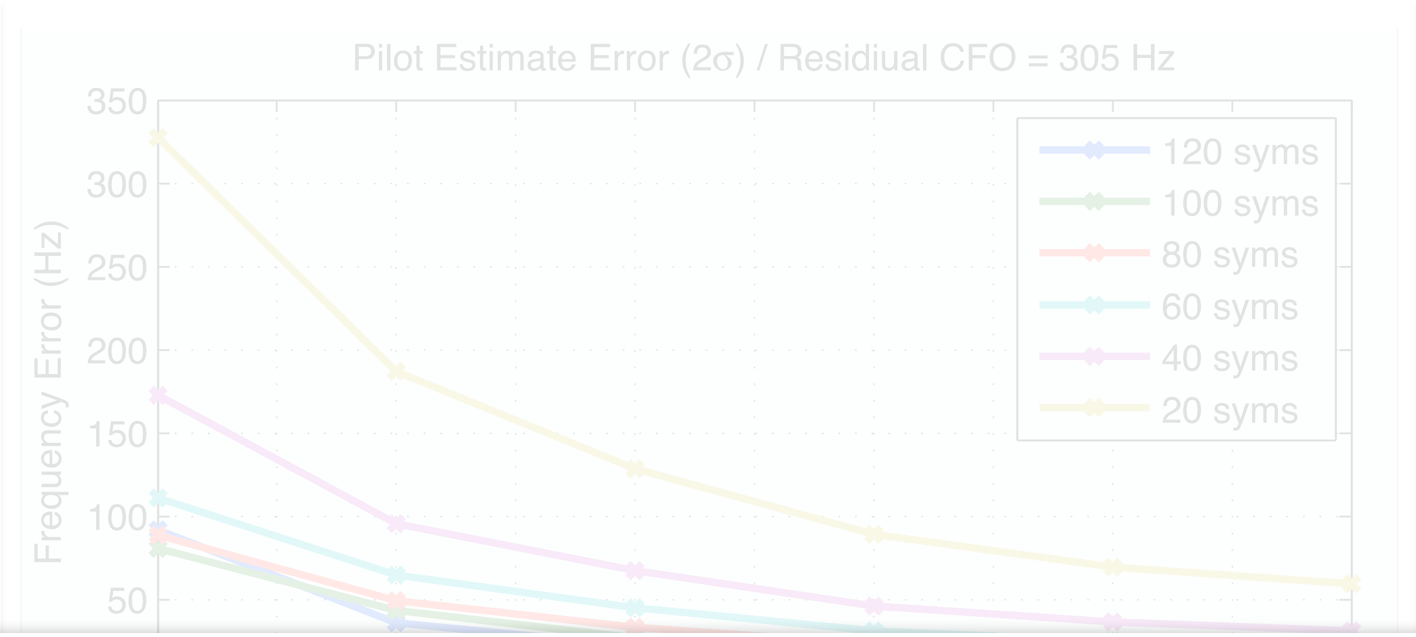


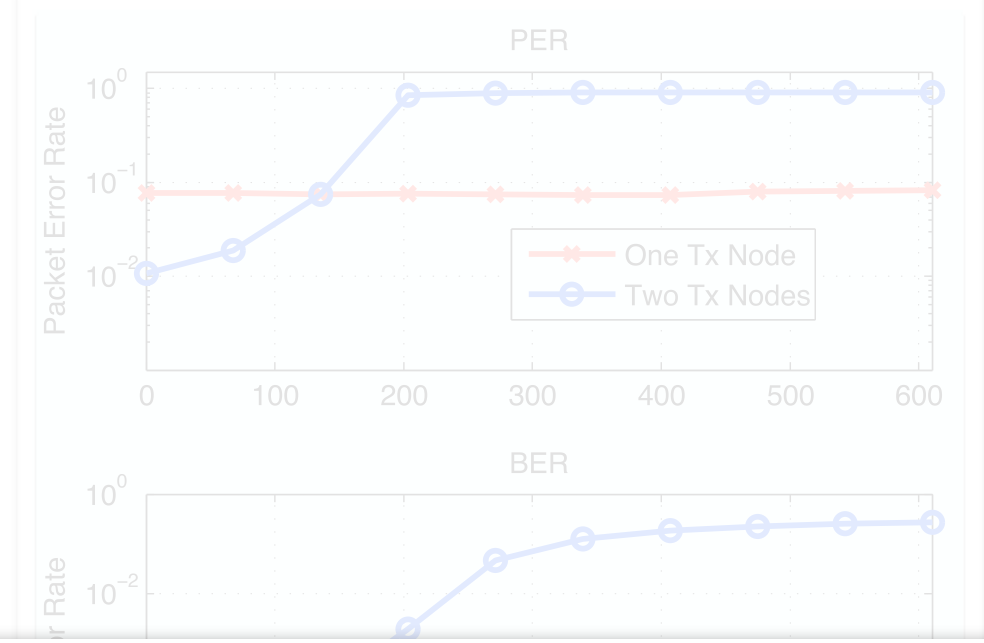






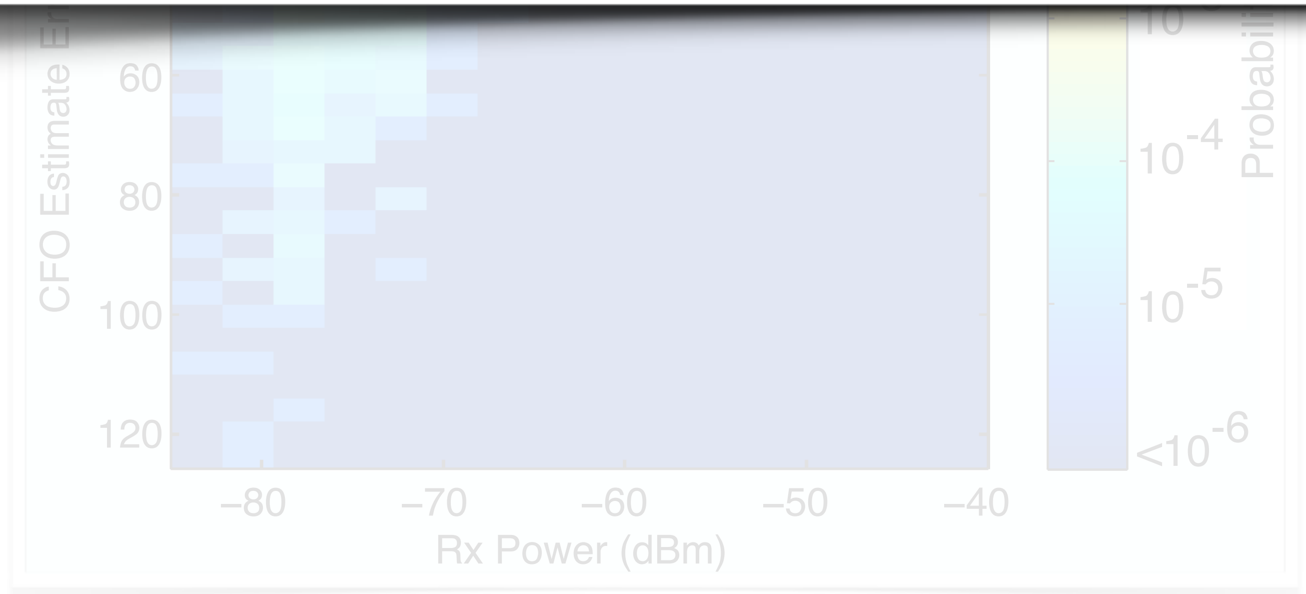
# Relay CFO Errors

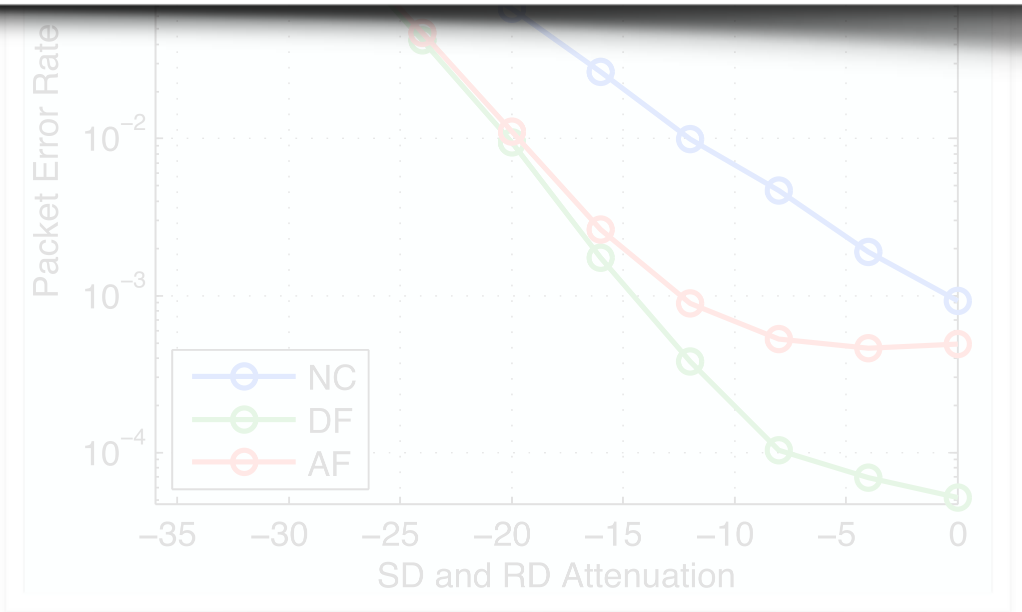




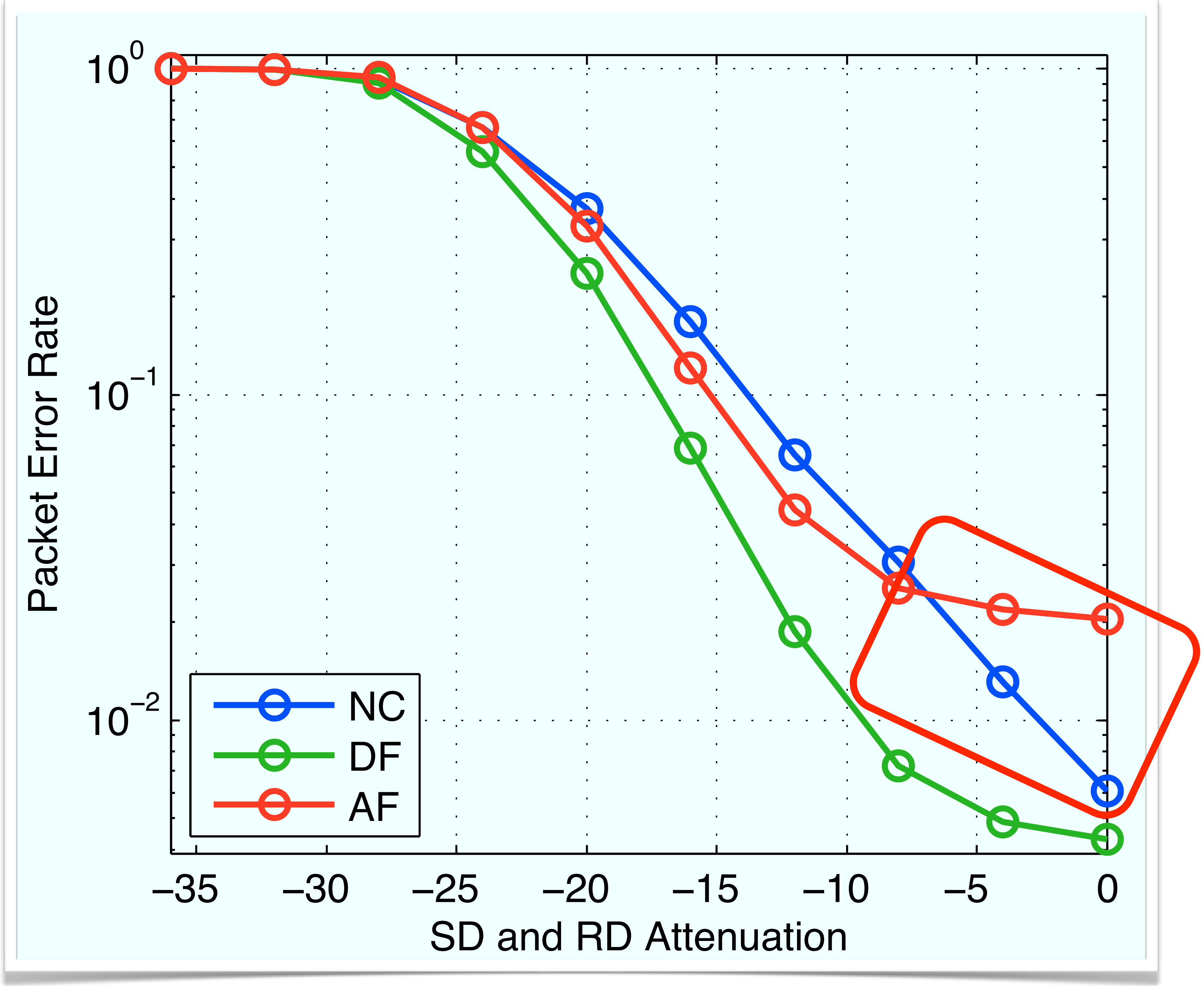
## Key Observations

1. Bad DF CFO estimates can cause errors.
2. These errors dominate only at high average SNR.





# Bit Error Densities

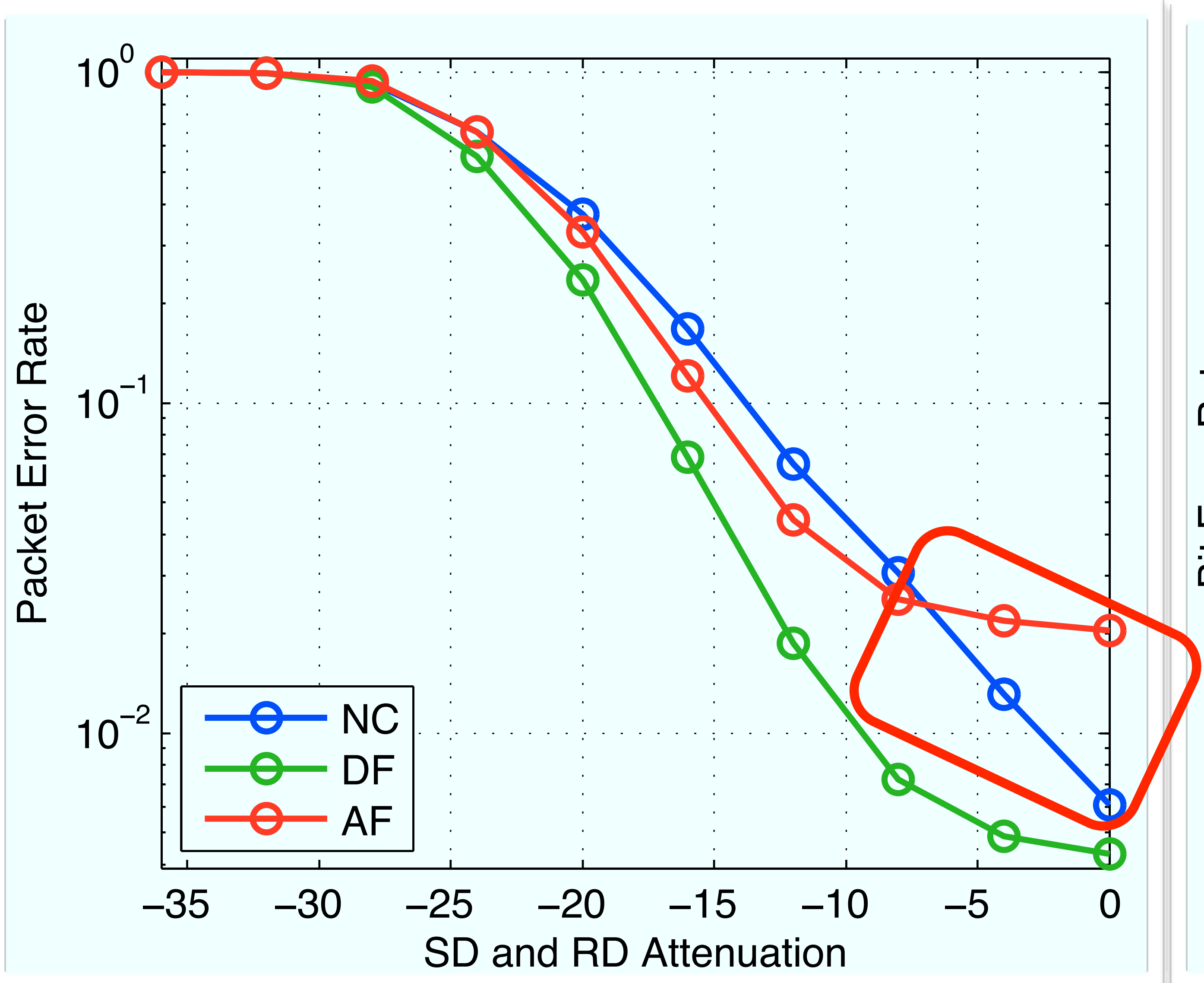


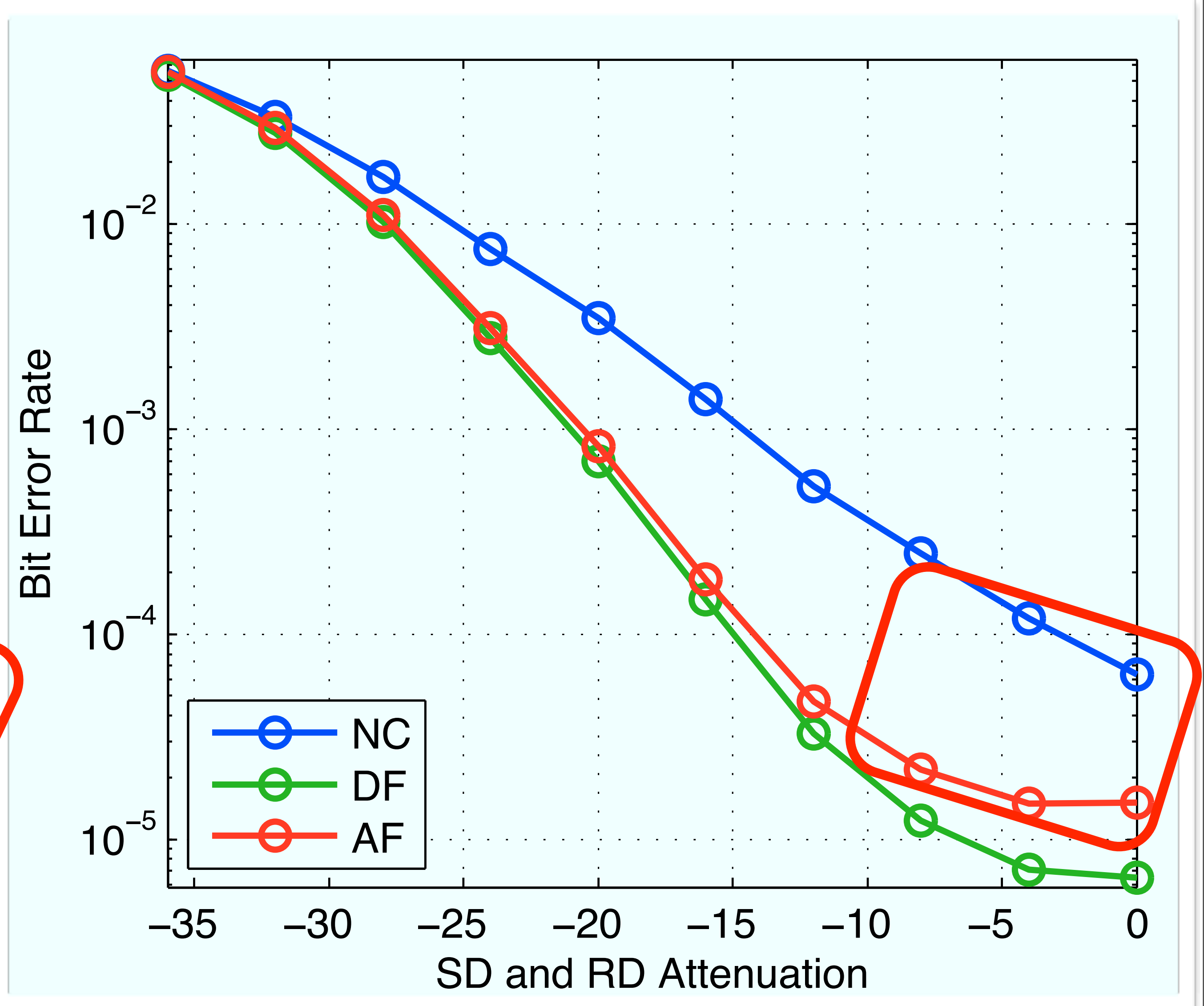
*1412 bytes / 16-QAM*

*Co-located S/R*

*Flat fading*

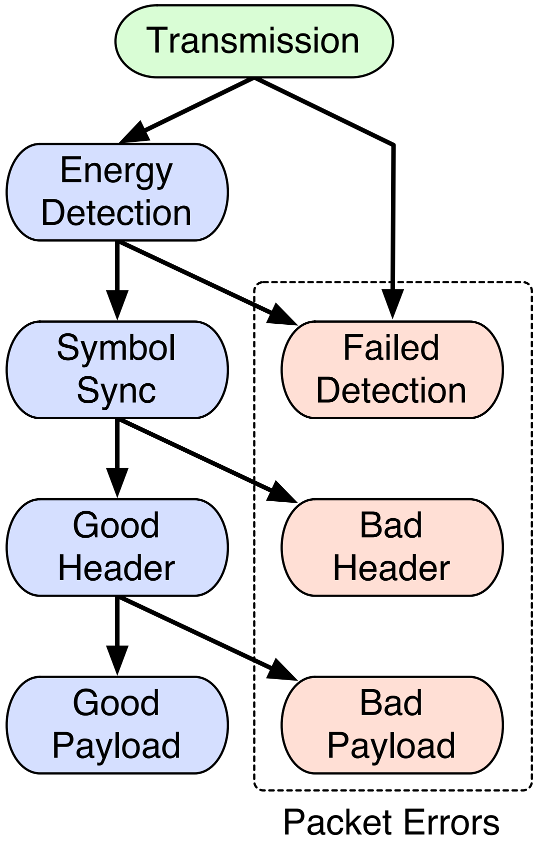
# Bit Error Densities





- PER / BER discrepancy for AF
- Noise amplification?

*1412 bytes / 16-QAM*
*Co-located S/R*
*Flat fading*



Transmission

Energy Detection

Symbol Sync

Good Header

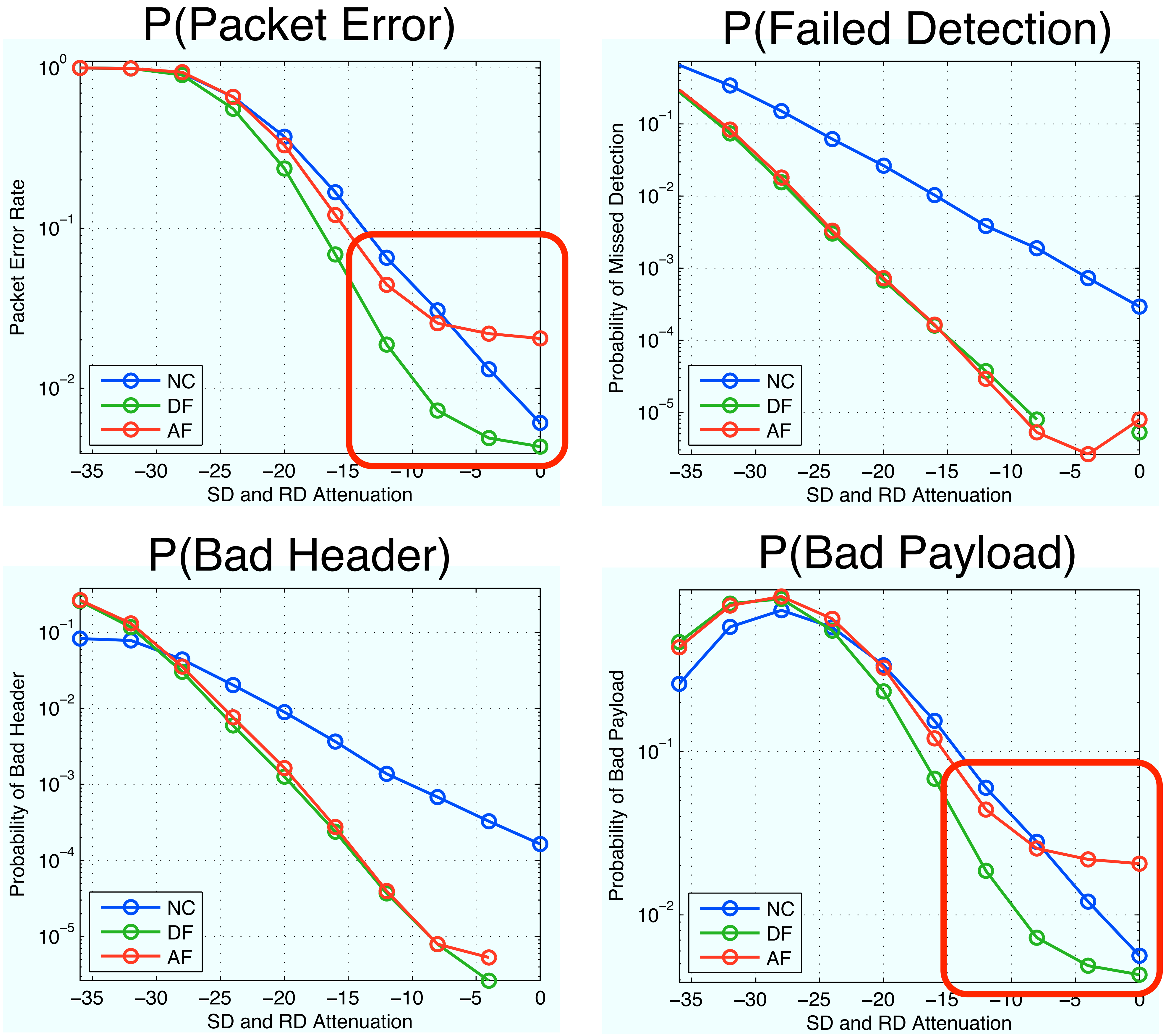
Good Payload

Failed Detection

Bad Header

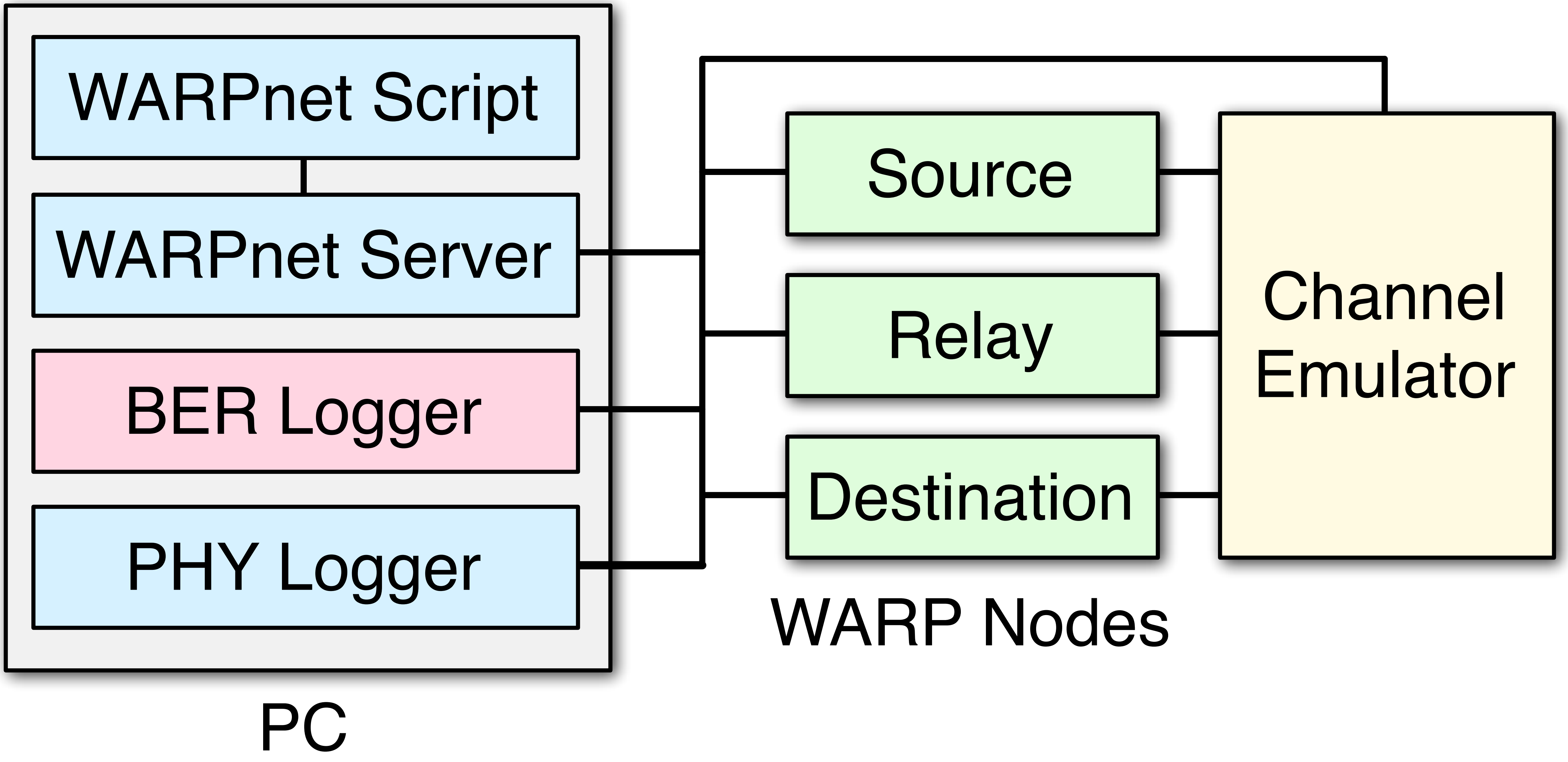
Bad Payload

Packet Errors



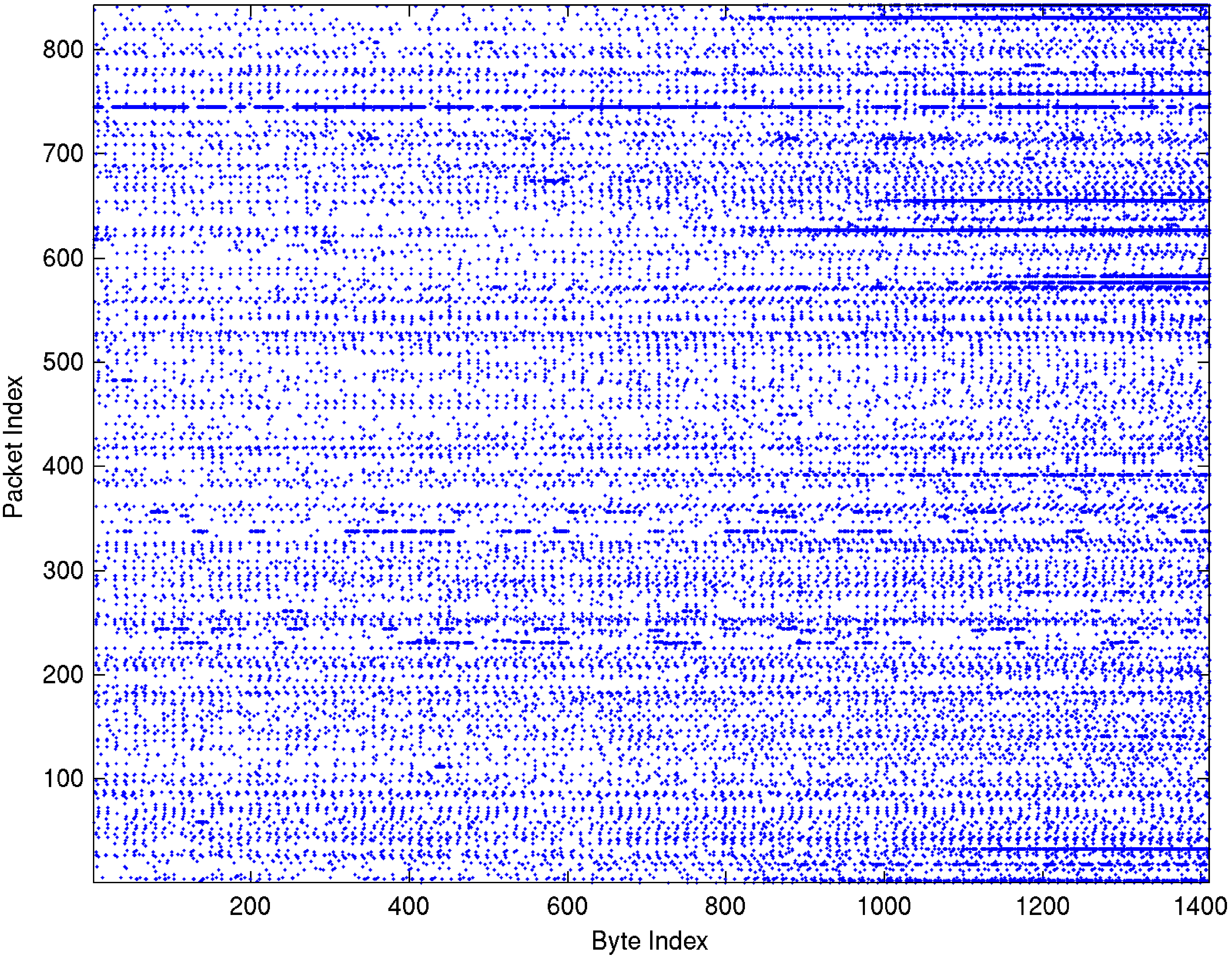
*1412 bytes / 16-QAM • Co-located S/R • Flat fading*

# WARPnet Framework

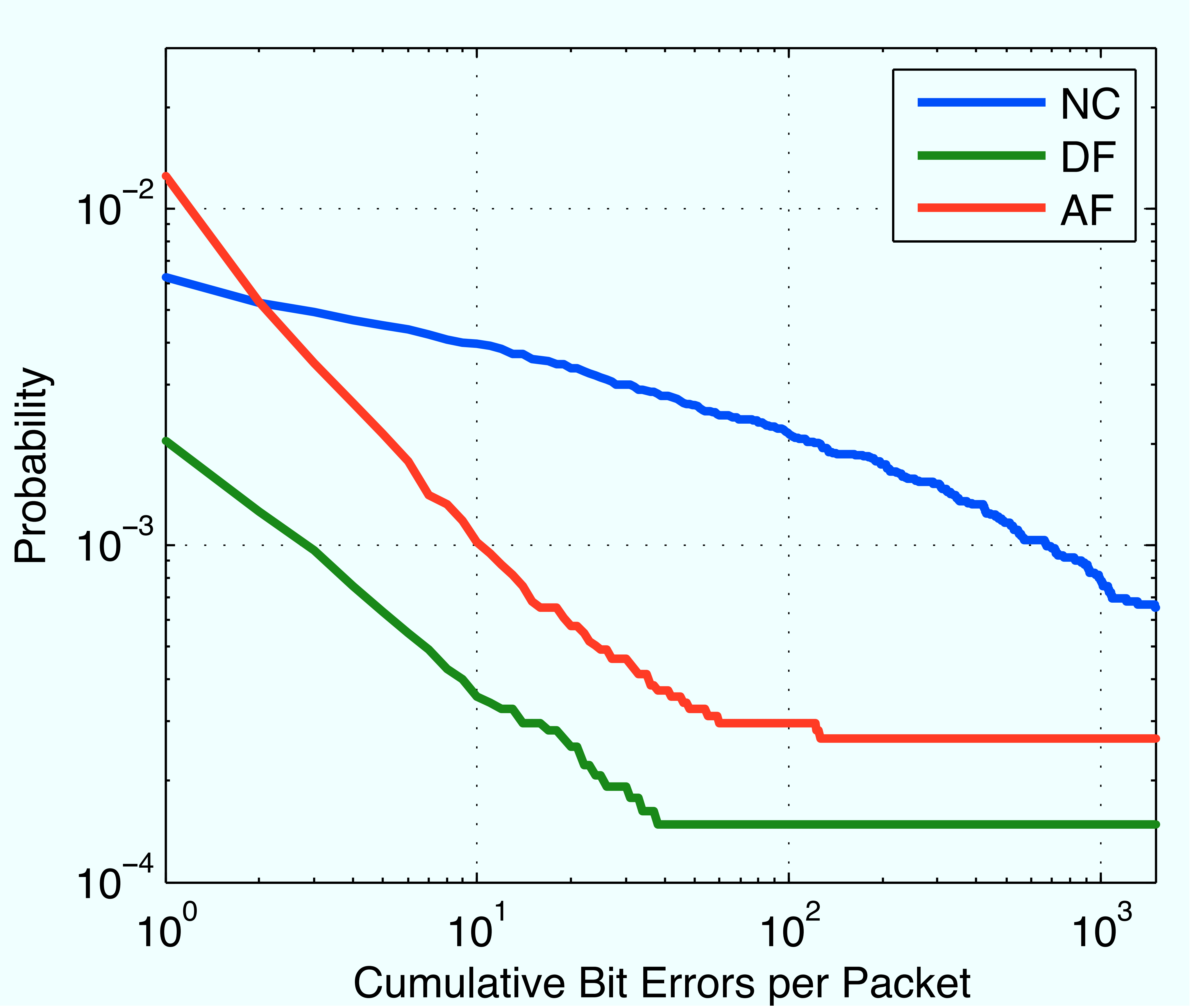


- BER Logger co-processor acts like node
- Logs per-packet bit error indices and MAC header

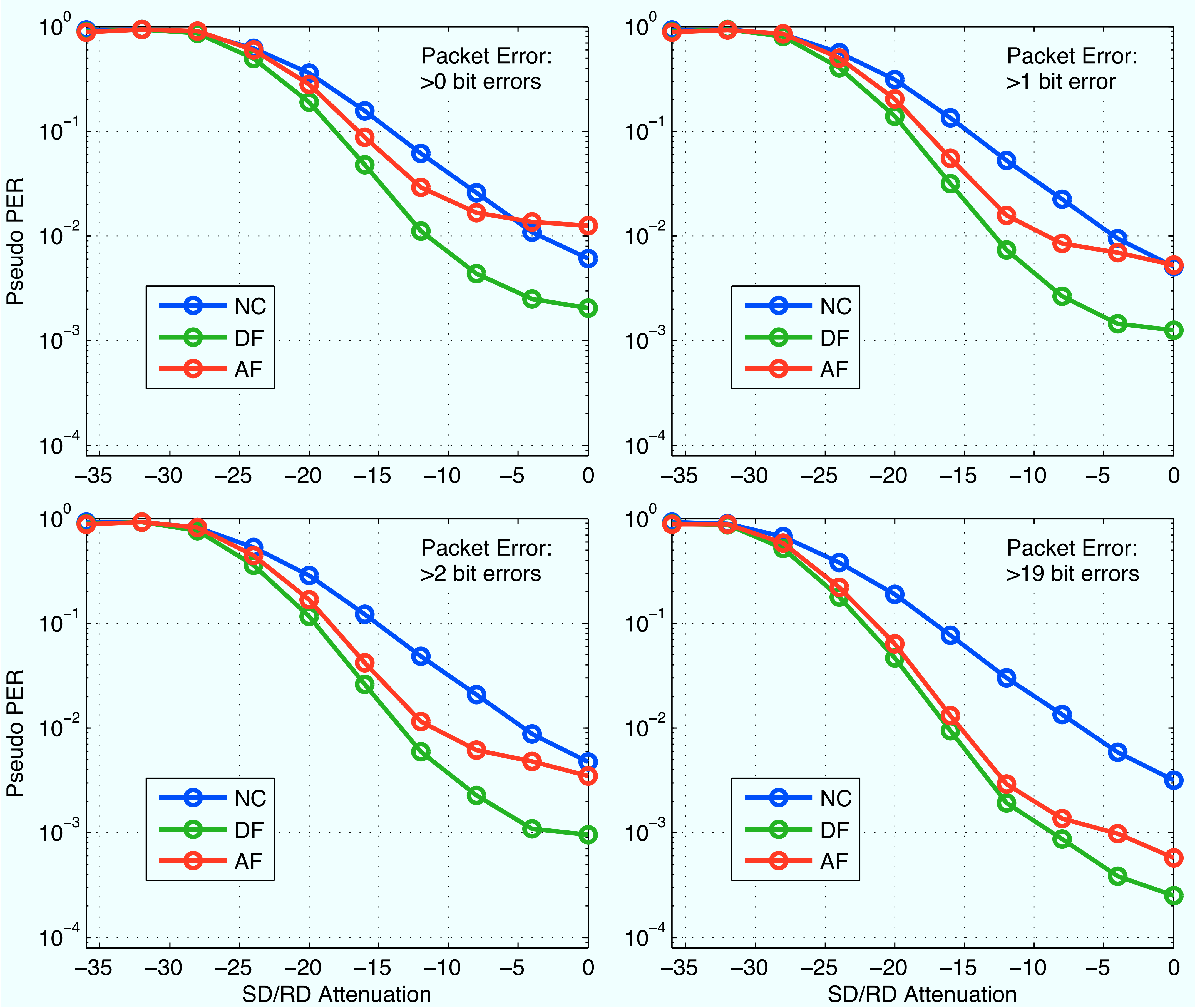
# Bit Error Densities



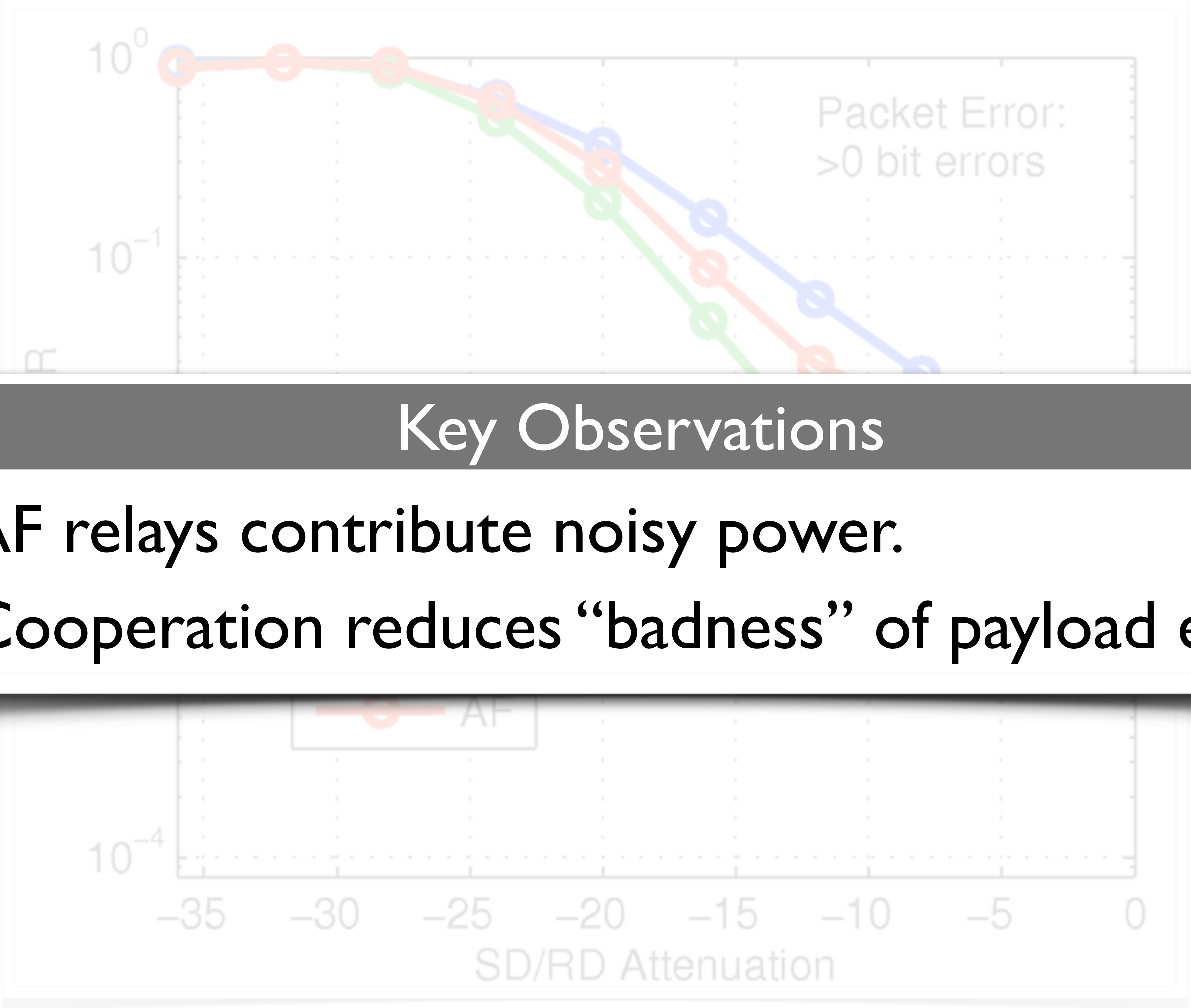
# Bit Error Densities



# Bit Error Densities



# Bit Error Densities



## Key Observations

1. AF relays contribute noisy power.
2. Cooperation reduces “badness” of payload errors.

# Results Summary

- Cooperative transceiver works
- Cooperation helps, sometimes a lot
- Real-world performance limitations
  - Our methodologies can isolate/explain them

# Future Work

- Transceiver extensions
- Cooperation in a network

# PHY Extensions

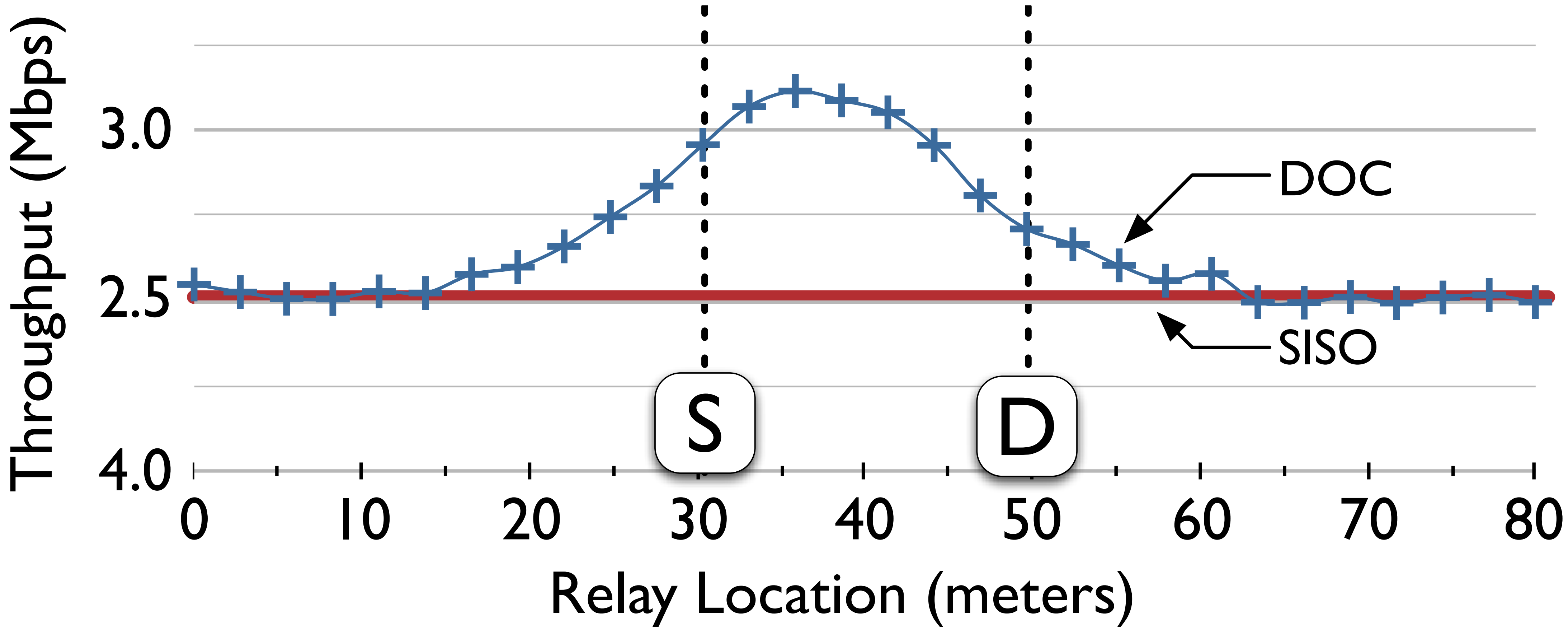
- Temporal combining
  - Extra frequency domain processing & storage
- Error correcting coding
  - Pseudo-PER framework will be useful
- Full duplex relaying
  - Recent narrowband results are promising
  - Extension to wideband/OFDM will be challenging

# Cooperation in a Network

- Physical layer characterization shows big gains
- Integration with higher layers is open question
  - How to select relays?
  - Preemptive or reactive cooperation?
  - How to balance local help vs. global hurt?
- Our PHY is network-ready
  - Early results are promising

# Distributed On-Demand Cooperation

- Collaboration with Chris Hunter
- Full PHY + MAC implementation
  - PHY cooperation for MAC re-transmissions
- Promising early results



# Conclusions

- Cooperation is possible in real systems
- Cooperation helps (sometimes a lot)
- Synchronization is hardest part
- Our designs are ready to enable future work

# Questions?